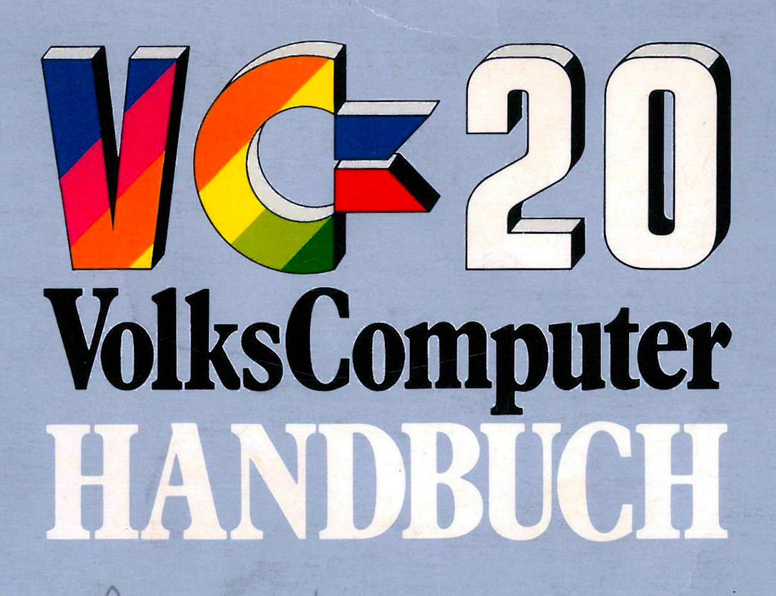


A = Zahl
A$ = Text



commodore

Commodore GmbH
Postfach 426
D-6078 Neu-Isenburg

Commodore AG Schweiz
Dufourstraße 9
CH-4010 Basel

Steiner Computer
Vertriebsgesellschaft mbH
Fleschgasse 2
A-1130 Wien

Änderungen vorbehalten.

PERSÖNLICHE DATENVERARBEITUNG MIT DEM



Ein freundliches Computer Handbuch

COMMODORE GMBH

VORWORT

Sie sind im Begriff einen freundlichen Computer kennenzulernen, freundlich im Preis, in der Größe, in der Benutzung, zum Lernen und Erfahrung sammeln. Sehr wichtig – Sie brauchen kein Programmierer zu sein, um ihn zu benutzen.

Falls Sie zum ersten Mal mit einem Computer arbeiten, wird Ihnen dieses Handbuch eine ausgezeichnete Einleitung in die Arbeit mit dem Computer geben. Im Gegensatz zu den meisten anderen Handbüchern brauchen Sie dieses Buch nicht vollständig durchzulesen um mit dem Gerät arbeiten zu können. Nachdem Sie Kapitel 1 gelesen haben, können Sie ein beliebiges Kapitel lesen, welches Sie interessiert. Falls Sie die Zeichentrickdarstellung interessiert, dann lesen Sie Kapitel 4; möchten Sie lieber Musik erzeugen, dann lesen Sie Kapitel 5.

Auf der ersten Seite jedes Kapitels steht ein Beispielprogramm. Geben Sie dieses Programm fehlerfrei ein, starten Sie es und beobachten Sie was passiert. Der Rest des Kapitels erklärt das Programm und zeigt noch weitere Möglichkeiten auf. Kapitel 7 enthält eine Zusammenstellung von wichtigen Programmtechniken und erklärt die Methoden, welche in den einzelnen Beispielprogrammen benutzt werden.

Sind Sie ein erfahrener Programmierer, so können Sie den VC-20 wie jeden anderen Mikrocomputer benutzen. Hilfreich ist die Vertrautheit mit anderen Commodore Computern, da das BASIC und die graphischen Zeichen nahezu identisch sind mit denen im CBM. Weiterführendes Nachschlagwerk und Programmierinformationen finden Sie im Anhang. Für anspruchsvollere Programmierung benutzen Sie das VC-20 Programmierhandbuch, welches Sie von Ihrem Händler beziehen können.

Haben Sie kein Interesse am Programmieren, so beachten Sie die ständig wachsende Bibliothek von Programm-Moduln und Programm-Bandkassetten. Die Moduln werden direkt in den VC-20 hineingesteckt und arbeiten automatisch. Die Programme auf Bandkassette werden mit Hilfe von dem VC 1530 Rekorder in den Computer eingelesen. Moduln und Bandkassetten beinhalten sowohl bestehende Spielprogramme wie „VC Invaders“, als auch Lernprogramme, die Ihnen helfen Aufgaben zu lösen und Berechnungen auszuführen.

Als Peripheriegeräte stehen Kassettenrekorder, Floppy-Disk und Drucker zur Verfügung, um nur einige zu nennen.

Computer werden in zunehmendem Maße Bestandteil unseres Alltags – zu Hause, in der Schule und bei der Arbeit. Jeder, der im Umgang mit Computern vertraut ist, wird in den kommenden Monaten und Jahren einen wichtigen Vorteil haben. Der VC-20 führt Sie nicht nur in die Welt der Computer ein, sondern gibt Ihnen auch die Grundlagen und die Anpassungsfähigkeit, die Sie brauchen um diese Welt zu erweitern.

Erfreuen Sie sich an Ihrer neuen Welt.

Inhaltsverzeichnis

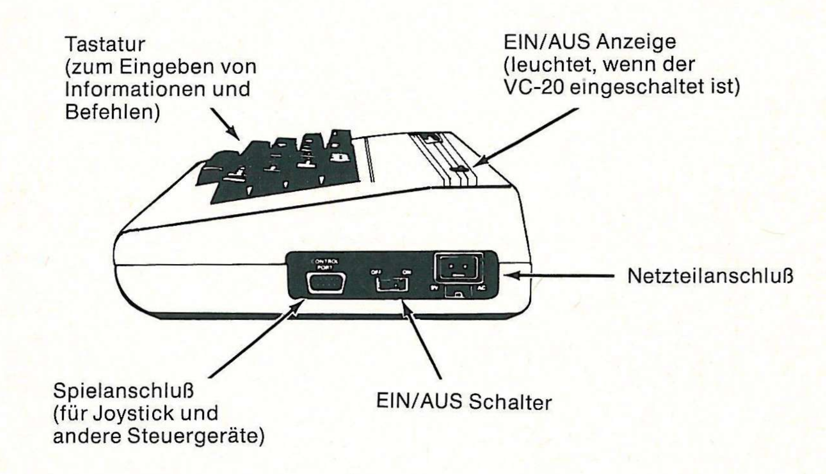
Kapitel **Seite**

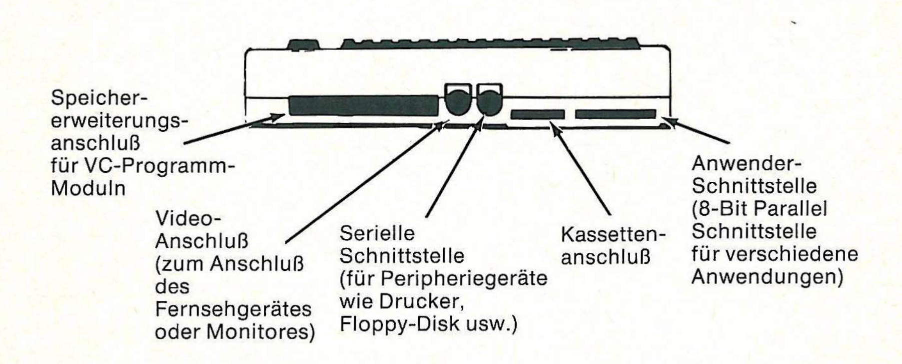
Anhang Seite

Inbetriebnahme und Anschluß des VC-20

Herzlich willkommen in der Datenverarbeitung. Die folgenden Anweisungen zeigen Ihnen Schritt für Schritt, wie Sie den VC-20 in Betrieb nehmen, an Ihr Fernsehgerät anschließen und seine einwandfreie Funktion überprüfen.

Zu Anfang werfen wir einen Blick auf den VC-20:



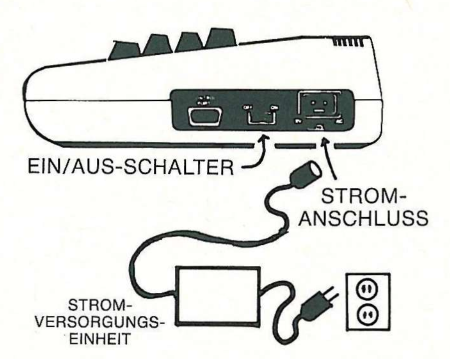


1. Wenn Sie die Verpackung öffnen, finden Sie folgenden Inhalt:
- VC-20-Volks-Computer,
- Stromversorgungs-Einheit (großes Gehäuse mit den Anschlußkabeln an die Steckdose und den VC-20),
- HF-Modulator (kleiner Metallkasten) und den Kabeln für VC-Videoausgang und FS-Antenneneingang.

2. Sie benötigen *2 Netzsteckdosen* eine für den VC-20 und eine weitere für das Bildschirmgerät (Fernsehgerät).

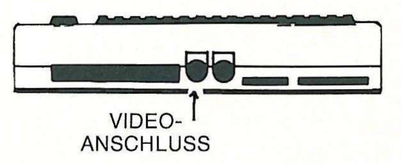
3. Achten Sie darauf, daß eine möglichst günstige Position der Geräte zueinander zusammengestellt wird, so daß diese leicht überblickbar und bequem zu bedienen sind.

4. Der Schalter zum *Ein- und Ausschalten* (ON/-OFF) des Gerätes befindet sich am rechten Teil des Tastaturgehäuses. Stellen Sie sicher, daß dieser Schalter auf OFF (Aus) gesetzt ist.

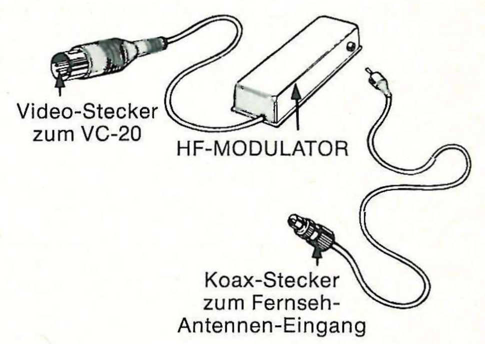


5. An der Stromversorgungs-Einheit sind 2 Kabel angebracht. Das eine Kabel ist für die *Netzstrom-Steckdose*, während das andere Kabel mit der am Gerät rechts befindlichen *Anschlußbuchse* zu verbinden ist. ANMERKUNG: Der Stromfluß zum Gerät ist erst dann unterbrochen, wenn der Stecker zum Netzteil herausgenommen wird. Bei Nichtbenutzung wird empfohlen, die Stromversorgung zu unterbrechen.

RÜCKANSICHT DES VC-20



6. Jetzt ist das *Video-Kabel* in den Video-Anschluß, der sich an der *Rückseite* des VC-20 befindet, und das andere Ende des Kabels mit dem *HF-Modulator* zu verbinden. ACHTUNG: Dieser Stecker paßt *nur* in den *Videoausgang* des VC-20 und *nicht* in den 6poligen seriellen Ausgang. Wenden Sie also keine Gewalt an.



7. Nun ist der *HF-Modulator* mit dem Fernsehgerät zu verbinden. Stecken Sie einfach das *Antennenkabel* (Koaxial-Kabel) in die *Antennen-Steckbuchse* Ihres Fernsehgeräts.

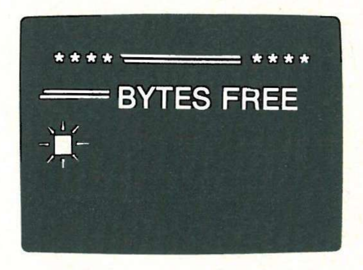
8. Jetzt kann das Fernsehgerät eingeschaltet werden.

ANMERKUNG: Sie können sich eine im Handel erhältliche Antennenabzweigung besorgen, damit Ihr VC-20 ständig betriebsbereit ist, und Sie zum Fernsehempfang die Kabel nicht umstecken müssen.

9. Einschalten des VC-20. (Die rote Leuchtanzeige, die sich oben auf dem Computer befindet, sollte aufleuchten.) Ist dies nicht der Fall, dann ist die in diesem Dokument beinhaltende Fehleraufklärungs-Tabelle zu lesen.

10. Wählen Sie nun an Ihrem Fernsehgerät einen freien Kanal (AV-Kanal) und stimmen Sie diesen auf dem Band III (UHF) mit der Frequenz des VC-20 ab.

Das hier zu sehende Bild sollte auf dem Bildschirm erscheinen, wobei dies u. U. einige Sekunden dauert. Ist das Bild immer noch nicht in Ordnung, oder fehlen die Farben, so schalten Sie den VC-20 einige Sekunden ab.



11. Die Farbeinstellung und -abstufung hängt selbstverständlich weitgehendst von den sich am Fernsehgerät befindenden Reglern ab. Je besser diese Einstellungsmöglichkeiten, umso besser ist die Farbe.

12. Sollten irgendwelche Probleme im Zusammenhang mit den hier beschriebenen Schritten auftauchen, ist die Fehleraufklärungs-Tabelle einzusehen.

Jetzt ist der VC-20 betriebsbereit

ANMERKUNG: Es ist möglich, anstatt eines *Fernsehgeräts* einen für die Computer-Bearbeitung geeigneten Bildschirm einzusetzen. In diesem Fall kann man den VC-20 direkt mit dem Monitor-Kabel verbinden, ohne den HF-Modulator einzusetzen.

Fehleraufklärungs-Tabelle

Symptom	Ursache	Fehlerbereinigung
KEIN BILD (Stromanzeige Aus)	VC-20 nicht eingeschaltet	Der Stromversorgungs-Schalter ist auf „ON" (Ein) zu setzen
	VC-20 nicht angeschlossen	Der sich direkt neben dem Stromversorgungsschalter befindende Strom-versorgungsstecker ist zu prüfen
	Stromverbindung unterbrochen	Prüfung des Anschlusses am Hauptnetz-Stecker
	Sicherung im VC-20 durchgebrannt	Ihr autorisierter COMMODORE-Händler ist zu benachrichtigen um die Sicherung auszutauschen.
KEIN BILD (Stromanzeige an)	(Schalten Sie den VC-20 einige Sekunden aus, bevor Sie neu starten)	
	Auf falschem TV-Kanal	Erneute Feinabstimmung (Band III, UHF)
	Falscher Anschluß	Überprüfen Sie nochmal alle Kabelverbindungen (Schalter dabei auf „OFF = Aus).
	Modulator nicht angeschlossen	Überprüfung des Anschlusses am Video-Anschluß
BILD OHNE FARBE:	Schlecht eingestelltes Fernsehgerät	Neueinstellung des Fernsehgeräts
	Ausschaltphase war zu kurz	Schalten Sie den VC-20 nochmal für einige Sekunden aus, bevor Sie wieder starten.
BILD MIT SCHLECHTER FARBABSTUFUNG	Schlechte Farb-einstellung beim Fernsehgerät („Bild ohne Farbe")	Neueinstellung von Farbabstufung, Helligkeit und Kontrast am Fernsehgerät
BILD MIT ÜBERMÄSSIG VIELEN HINTERGRUND-GERÄUSCHEN	Fernseh-Lautstärke zu hoch eingestellt (siehe „Bild ohne Farbe")	Neueinstellung der Lautstärke des Fernseh-gerätes
BILD IN ORDNUNG, JEDOCH KEIN TON	Lautstärke des Fernseh-gerätes zu niedrig eingestellt	Neueinstellung der Lautstärke des Fernseh-gerätes

* Bei VC-20 wird eine Sicherung von 3 Ampere (TRÄGE) eingesetzt.



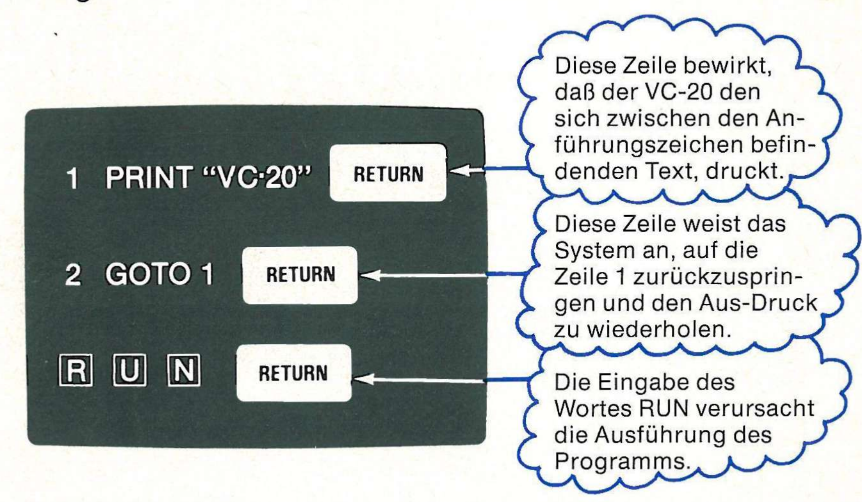


Lernen Sie den Umgang mit dem VC-20 kennen

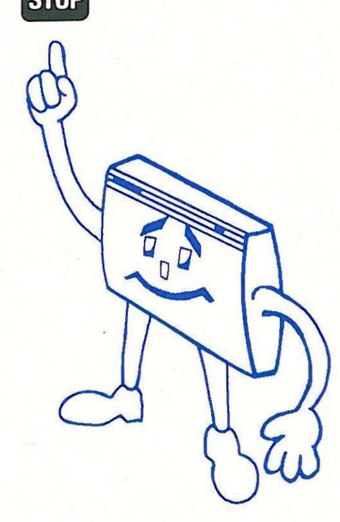
- Der Anfang – Kleine Experimente
- Das erste Programm

Programmvorschlag

Bringen Sie jetzt den VC-20 mit folgendem Mini-Programm zum Laufen!

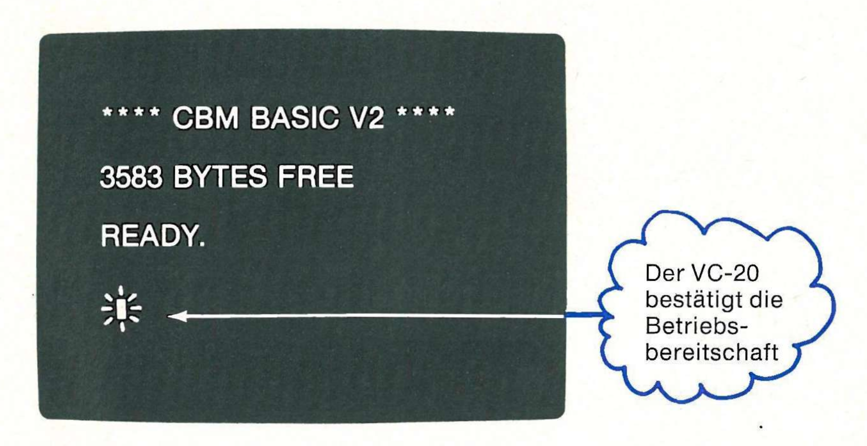


Um das Programm in seiner Ausführung anzuhalten, drücken Sie die RUN STOP -Taste.



Der Anfang – Kleine Experimente

NA ALSO – das System läuft! Der VC-20 wartet jetzt nur darauf, die nächsten Befehle entgegenzunehmen. Das dunkelblaue, blinkende Rechteck *(der Cursor)* ist das sichtbare Zeichen für die Betriebsbereitschaft.



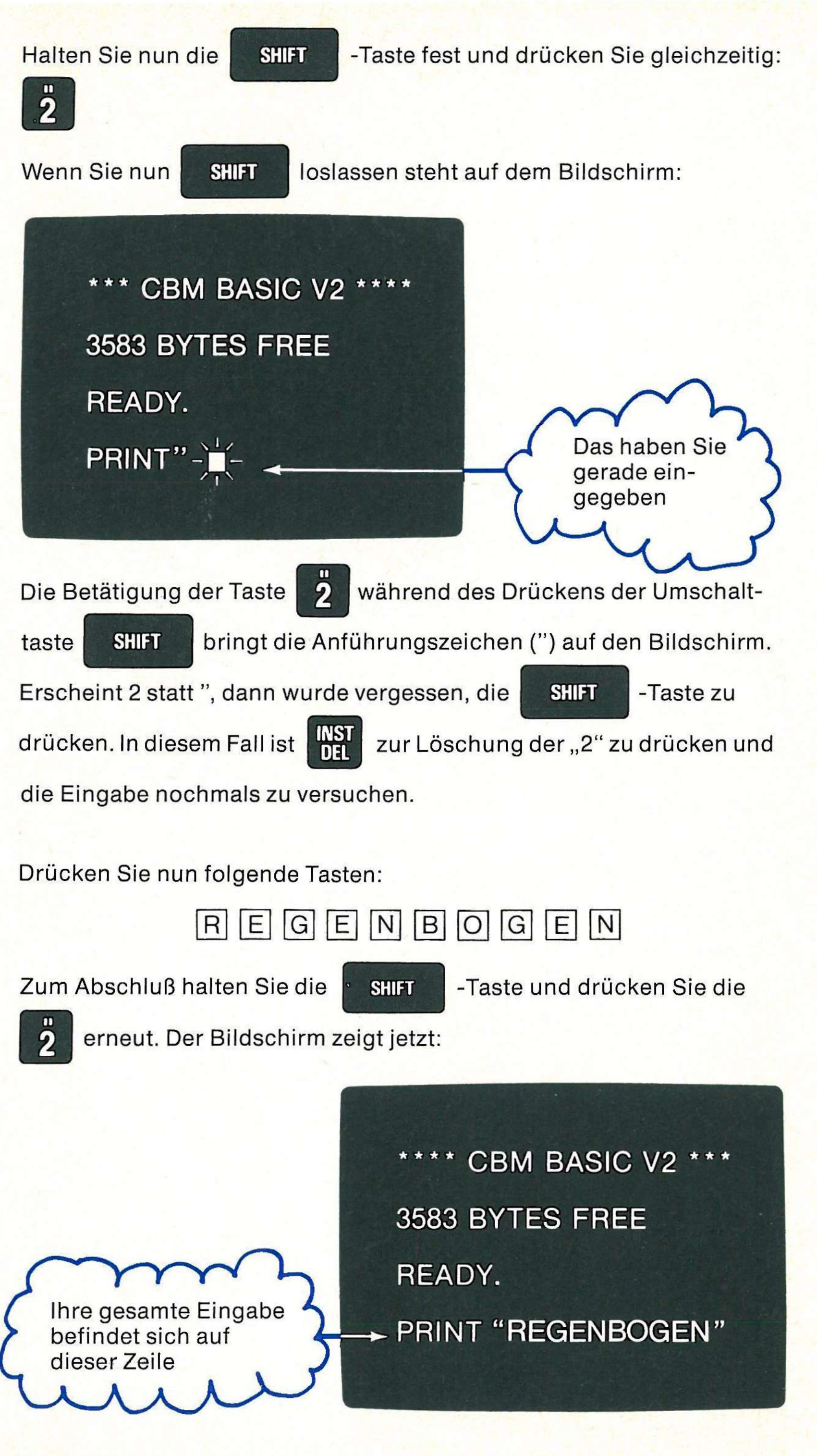
Hier ein VC-20 Tip:

Haben Sie ein unerwünschtes Zeichen gedrückt, dann können Sie dies mit der INST DEL -Taste rückgängig machen. Bei jedem Druck auf diese Taste löschen Sie das links neben dem „Cursor" befindliche Zeichen. Das können Sie so oft wie nötig wiederholen.

Weiter im Fahrplan! Drücken Sie nun folgende Tasten:

[P] [R] [I] [N] [T]

Sie sehen, wie der Cursor bei jeder Betätigung einer Taste um eine Stelle nach rechts rückt. Der Cursor sagt uns also, an welcher Stelle das nächste Zeichen auf dem Bildschirm erscheint. Als nächstes kommt die Umschalt-Taste (SHIFT) dran. Davon sind zwei vorhanden, welche folgendermaßen aussehen: SHIFT



Halten Sie nun die SHIFT -Taste fest und drücken Sie gleichzeitig: 2

Wenn Sie nun SHIFT loslassen steht auf dem Bildschirm:

```
*** CBM BASIC V2 ****
3583 BYTES FREE
READY.
PRINT"■
```


Die Betätigung der Taste 2 während des Drückens der Umschalttaste SHIFT bringt die Anführungszeichen (") auf den Bildschirm. Erscheint 2 statt ", dann wurde vergessen, die SHIFT -Taste zu drücken. In diesem Fall ist INST DEL zur Löschung der „2“ zu drücken und die Eingabe nochmals zu versuchen.

Drücken Sie nun folgende Tasten:

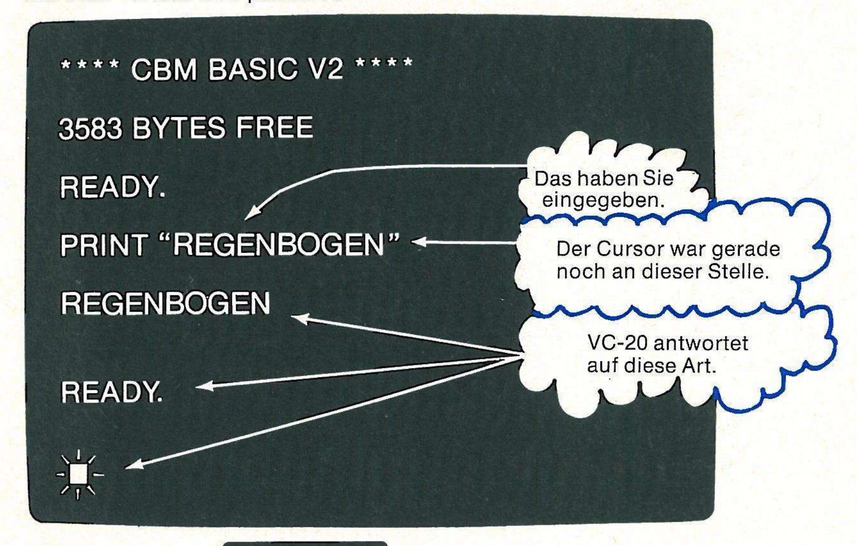
R E G E N B O G E N

Zum Abschluß halten Sie die SHIFT -Taste und drücken Sie die 2 erneut. Der Bildschirm zeigt jetzt:

```
**** CBM BASIC V2 ***
3583 BYTES FREE
READY.
PRINT "REGENBOGEN"
```

Jetzt wenden wir uns der RETURN -Taste zu (bei einer elektrischen Schreibmaschine wäre dies die „Wagenrücklauf"-Taste).

Drücken Sie nun die RETURN -Taste und beobachten Sie, was auf dem Bildschirm passiert.

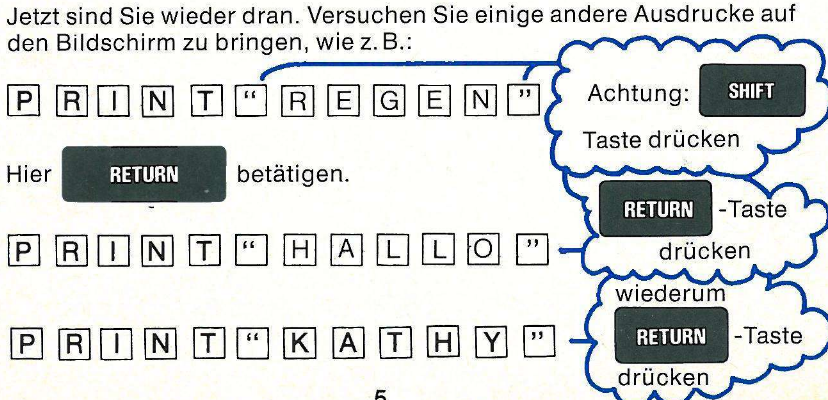


Das Drücken von RETURN bedeutet für den VC-20 die Beendigung der Eingabe.

Der VC-20 sieht sich unmittelbar danach an, welcher Art die eingegebenen Informationen sind. Er erkennt, daß er angewiesen wurde etwas auszuführen (in diesem Fall *„Drucken"*). Der VC-20 „druckt" *(PRINT)* alles, was sich zwischen den Anführungsstrichen befindet (REGENBOGEN).

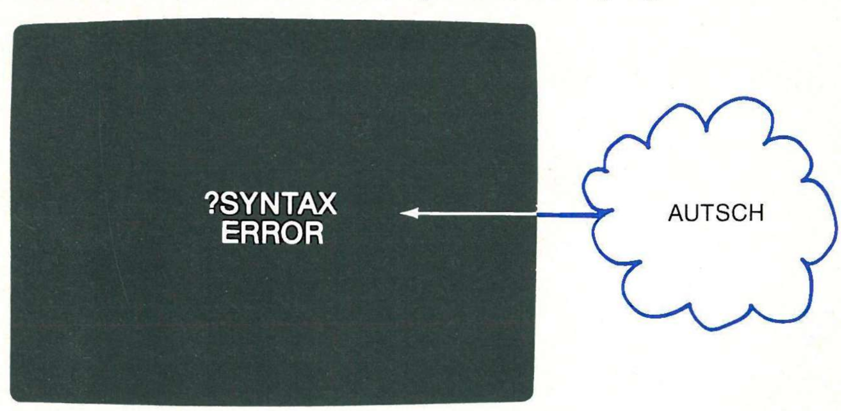
Hat der VC-20 den „Druck" des Wortes REGENBOGEN beendet, teilt er dies durch die Nachricht READY mit und der blinkende Cursor erscheint wieder.

Jetzt sind Sie wieder dran. Versuchen Sie einige andere Ausdrucke auf den Bildschirm zu bringen, wie z. B.:



Versuchen Sie jetzt mal Tasten zwischen den Anführungszeichen zu drücken, die nicht Buchstaben sind. Der VC-20 stört sich nicht daran.

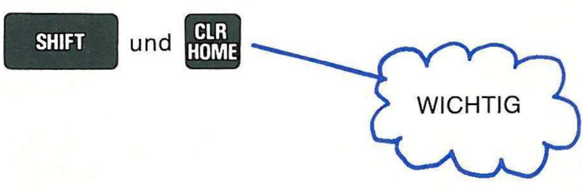
ABER: Sollten Sie das Wort „PRINT" falsch geschrieben haben, dann wird Ihnen der VC-20 dies mit der folgenden Meldung sagen:



Macht nichts! *Haben Sie keine Angst, daß sich der VC-20 durch irgend eine Eingabe verletzt fühlen könnte* (es sei denn, daß Sie allzu rabiat und gewaltsam mit der Tastatur umgehen). Sollten Sie jedoch mal einen Fehler gemacht haben, so wird Ihnen dies der VC-20 durch eine Fehlermeldung zeigen. Lassen Sie sich also durch einen „Syntax Error" nicht von weiteren Experimenten abhalten.

In kurzer Zeit wird der Bildschirm mit all den Eingaben vollgepfropft sein. Bei dem VC-20 gibt es eine bequeme Möglichkeit, sich der bisher eingegebenen Informationen vollständig zu entledigen, indem die CLR-Taste in Verbindung mit der Umschalt-Taste (SHIFT) betätigt wird.

Dazu ist die folgende Tastenkombination zu drücken:



Direkt danach ist der Bildschirm wie leergefegt. Alles, was Sie oder der VC-20 auf den Bildschirm gebracht haben, verschwindet, nur der „Cursor" blinkt noch im linken oberen Eck.

Merken Sie sich diese Möglichkeit gut, denn gerade am Anfang werden Sie sie häufig benutzen.

Das erste Programm

Hat sich der VC-20 so verhalten, wie Sie es sich wünschen – ganz egal, wie bizarr die Eingaben und die Antworten auch waren – dann ist er auch bereit, jede andere „Schandtat" (fast jede) zu begehen. Als nächstes sollten Sie sich vornehmen, *das erste „Programm"* einzugeben.

Schritt 1: Löschen des gesamten Bildschirminhaltes, indem Sie die Tastenkombination SHIFT (Umschalttaste) und CLR HOME gleichzeitig betätigen.

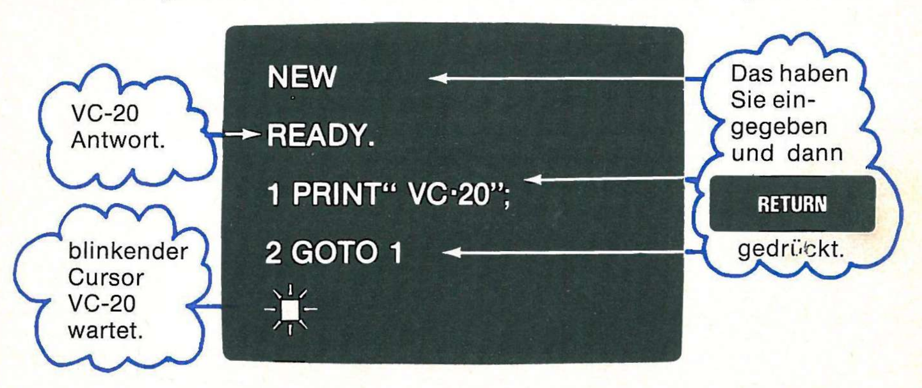
Schritt 2: Eingabe von N E W und Drücken der RETURN -Taste.

Schritt 3: Eingabe: 1 P R I N T " SPACE V C – 2 0 " ; (Strichpunkt unter Verwendung der SHIFT -Taste) und Betätigung der RETURN -Taste

Schritt 4: Eingabe 2 G O T O 1 und RETURN drücken.

Hierzu sind nicht die Buchstaben dieses Wortes einzugeben. Es ist lediglich die Leertaste bzw. Leerleiste zu drücken.

Nach Eingabe dieser Information sieht der Bildschirm wie folgt aus:



VC-20-TIP: Korrektur von Fehlern innerhalb eines Programmes

Für eine fehlerhafte Eingabe stehen die folgenden Korrektur-Möglichkeiten zur Verfügung:

1. *Es ist jederzeit möglich, eine Zeile neu einzugeben,* wonach vom VC-20 die alte Zeile durch die neue ersetzt wird. Wurde z. B. das folgende Programm eingegeben:

   10 PRINN „VC-20" — FEHLER

   20 GOTO 10

   dann ist es möglich, durch das wiederholte Drücken der RETURN -Taste einige Zeilen nach unten zu springen und den Fehler durch Eingabe der folgenden Zeile zu korrigieren:

   10 PRINT „VC-20" und RETURN

   Direkt danach ersetzt der VC-20 die alte Zeile des Programms mit der neuen und das Programm ist lauffähig. Überzeugen Sie sich durch L I S T RETURN . Das Ersetzen von Zeilen innerhalb eines Programmes ist auch ein schneller und problemloser Weg während der Experimentierphase.

2. *Es ist möglich, eine unerwünschte Zeile zu entfernen,* indem die *Nummer* dieser Zeile eingegeben wird, danach ist die RETURN -Taste zu drücken. Die vollständige Zeile wird gelöscht.

3. *Es ist möglich, die sich innerhalb einer Zeile befindenden Fehler zu korrigieren.* Setzen Sie den „Cursor" mit den Cursor-Steuertasten an die fehlerhafte Stelle innerhalb der Programmzeile und korrigieren Sie diese. Bestätigen Sie anschließend mit der RETURN -Taste.

   Korrekturen innerhalb von Anführungszeichen können dabei jedoch einige Schwierigkeiten bereiten. Überschreiben Sie einfachheitshalber solche fehlerhaften Zeilen von Anfang an.

4. Die INST-Taste (aktiviert durch SHIFT -Taste und INST DEL -Taste) ermöglicht es, Zeichen den schon eingegebenen hinzuzufügen, indem Platz an der gewünschten Stelle zur Verfügung gestellt wird, an welchem neue Zeichen eingegeben werden können.

5. Die DELETE-Taste (es ist lediglich INST DEL zu betätigen): erlaubt es, die sich direkt links vom Cursor befindenden Zeichen zu löschen.

Sieht alles richtig aus, kann jetzt der folgende Befehl eingegeben

RETURN :

und danach R U N RETURN gedrückt werden.

Der Bildschirm sollte sich jetzt mit „VC-20" auffüllen. Von Zeit zu Zeit wird es so aussehen, als ob es sich hierbei um Zeichen handelt, die sich im Zeichentrick-Verfahren nach oben bewegen.

```
VC-20VC-20VC-20VC-20VC-20VC
C-20VC-20VC-20VC-20VC-20VC-2
-20VC-20VC-20VC-20VC-20VC-20
20VC-20VC-20VC-20VC-20VC-20V
0VC-20VC-20VC-20VC-20VC-20V
VC-20VC-20VC-20VC-20VC-20VC
C-20VC-20VC-20VC-20VC-20VC-2
-20VC-20VC-20VC-20VC-20VC-20
20VC-20VC-20VC-20VC-20VC-20V
VC-20VC-20VC-20VC-20VC-20VC
VC-20VC-20VC-20VC-20VC-20VC
VC-20VC-20VC-20VC-20VC-20VC
```

Wenn Sie wollen, können Sie die Geschwindigkeit abbremsen, indem die Taste CTRL gedrückt wird.

So lange die CTRL -Taste heruntergedrückt bleibt, wird auch die Verarbeitungs-Geschwindigkeit *des Programms gebremst.*

Erstaunlich! Ihr VC-20 ist voll von phantastischen Einrichungen. Durch das Drücken einer einzelnen Taste können Sie die Geschwindigkeit des VC-20 drosseln, mit der er Zeichen auf dem Bildschirm „druckt".

So weit in Ordnung! Aber wie kann man das Programm jedoch anhalten?

Eine gute Frage. Dafür ist die Taste 

zu finden.

Durch Drücken dieser Taste wird das Programm angehalten und die folgende Nachricht erscheint:

BREAK IN 1

READY

(Keine Angst – Ihr VC-20 wurde nicht beschädigt. „BREAK" bedeutet in der VC-20-Sprache lediglich, daß der Verarbeitungsvorgang angehalten wurde.) Nun taucht auch der Cursor wieder auf. Haben Sie eigentlich bemerkt, daß er die ganze Zeit, während das Programm lief, nicht zu sehen war?

Jetzt sollten Sie einmal feststellen, ob sich das Programm noch im Speicher befindet. Schreiben Sie einfach:





Das Programm (Zeilen 1 und 2) wird auf dem Bildschirm erscheinen. Nach erneuter Eingabe des Befehls RUN wird das Programm wieder weiterlaufen.

Bis jetzt wurden Sie mit einer ganzen Reihe von Bedienungsmöglichkeiten des VC-20 vertraut gemacht, die Sie auch in den folgenden Kapiteln verwenden werden. Es wurden beschrieben:

- „Drucken“ (PRINT) von Zeichen auf dem Bildschirm;
- Löschen des Bildschirminhalts (SHIFT und CLR);
- Schreiben des ersten VC-20-Programms und die Erzeugung eines bewegenden Bildes;
- Reduzierung der Verarbeitungs-Geschwindigkeit (CTRL) eines ausführenden Programms;
- Anhalten der Ausführung des Programms unter Verwendung der STOP-Taste (RUN/STOP-Taste);
- Auflistung (LIST) eines Programms;
- einfache Korrektur von fehlerhaften Eingaben.

Bei der weiteren genaueren Betrachtung der sich in diesem Handbuch befindenden Kapitel entdecken Sie eine Vielfalt der hier beschriebenen Möglichkeiten wieder. Sie sollten sich jedoch nicht daran stören, wenn zu diesem Zeitpunkt noch einige Punkte offen sind. *Experimentieren* Sie einfach weiter und die meisten Ihrer Fragen lösen sich dabei wie von selbst.

Dieses Handbuch ist so konzipiert und angelegt, daß Sie sich mit *jedem Kapitel* einzeln befassen können, welches für Sie interessant erscheint. Es ist absolut nicht notwendig, jedes einzelne Kapitel der Reihe nach durchzulesen, um sich mit dem VC-20 vertraut zu machen. Es ist jedoch zu empfehlen, daß ein Kapitel, welches von Interesse erscheint, vom Anfang an durchgelesen wird. Wir sind der Meinung, daß jedes einzelne Thema schonend und trotzdem ausführlich behandelt wird und somit die Lerneffizienz am größten sein wird.

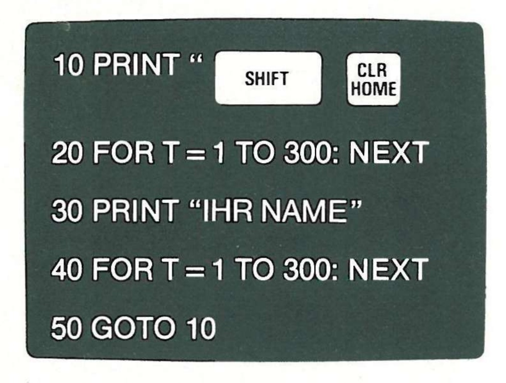


Benutzung von Bildschirm und Tastatur

- Das erste graphische Zeichen
- Die Bildschirmgröße und -Kapazität
- Eine Reise über die VC-20-Tastatur
- „Druck“ auf dem Bildschirm
- Der VC-20-Rechner
- Einführung in die Farbgestaltung

Programmvorschlag

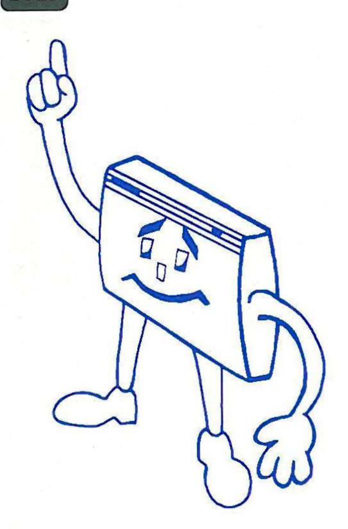
Bringen Sie den VC-20 mit folgendem Mini-Programm zum Laufen!



```
10 PRINT " SHIFT CLR/HOME
20 FOR T = 1 TO 300: NEXT
30 PRINT "IHR NAME"
40 FOR T = 1 TO 300: NEXT
50 GOTO 10
```

Eingabe: R U N und Drücken der Taste RETURN

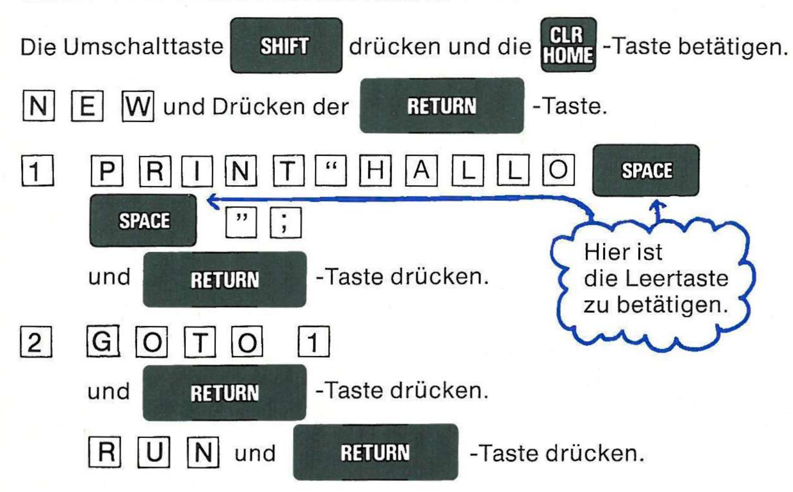
Um das Programm in der Ausführung anzuhalten, ist die RUN STOP -Taste zu betätigen.



Benutzung des Bildschirms und der Tastatur

Die Voraussetzung zum Verständnis dieses Kapitels ist, daß Sie Kapitel 1 *eingehend studiert haben.* Sollte das nicht der Fall sein, empfehlen wir, zumindest die letzten zwei Abschnitte dieses Kapitels zu lesen, da diese erklären, wie unter Verwendung der Tastatur die Steuerung der Ausgabe (PRINT) auf dem Bildschirm erfolgt.

Geben Sie am Anfang die folgenden Informationen ein. Vergessen Sie dabei nicht die Zeilennummern und Kommas.

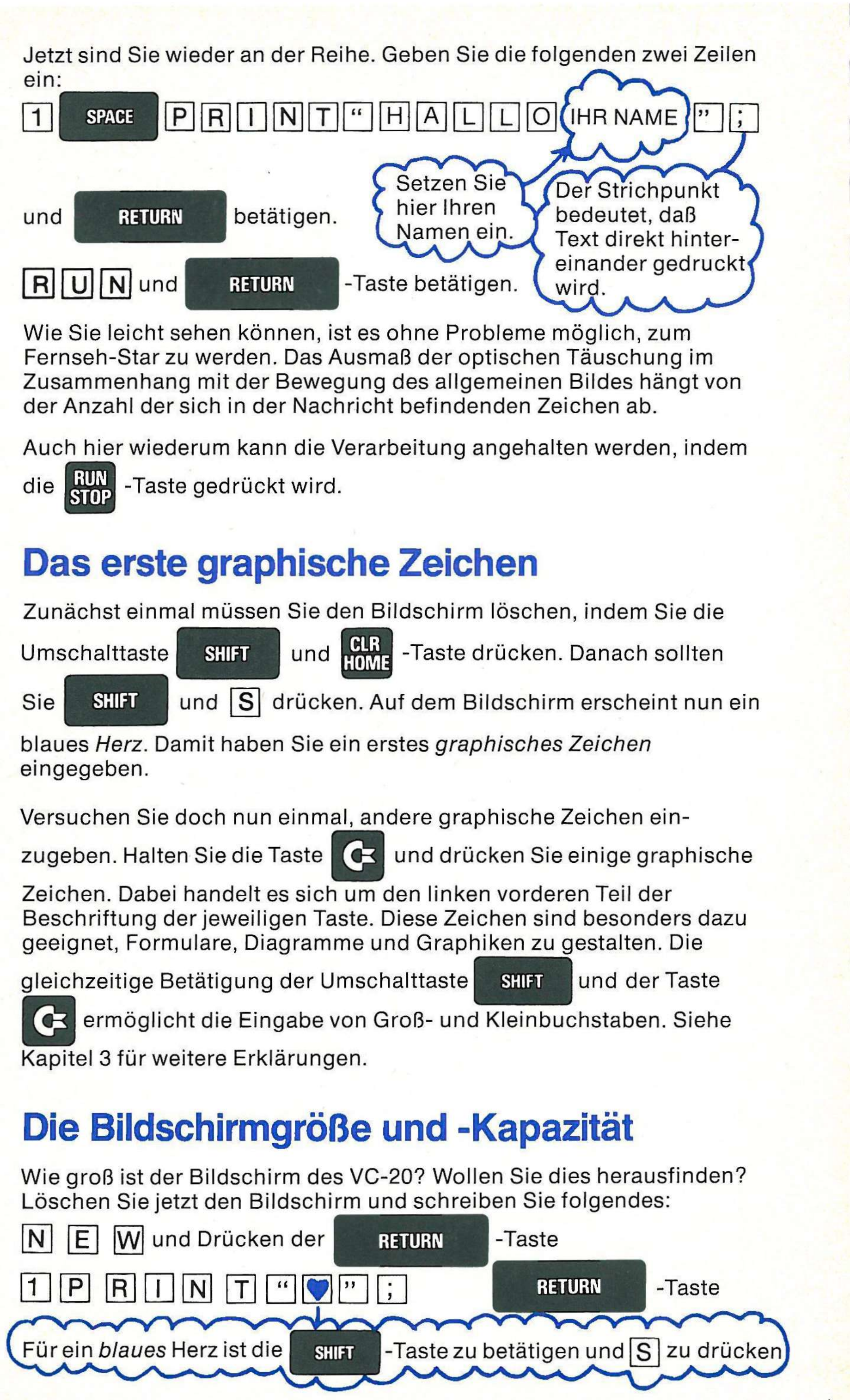


Wenn Sie nun RUN eingegeben, dann sollte sich der Bildschirm mit dem folgenden Wort auffüllen:

HALLO

Es hat den Anschein, als ob sich die Wörter nach oben und zur Seite bewegen. Durch die Betätigung der CTRL-Taste kann die Verarbeitungsgeschwindigkeit reduziert werden. VC-20 „druckt“ (PRINT) die eingegebene Nachricht im unteren Teil des Bildschirms. Nachdem der Bildschirm total aufgefüllt ist, werden die Zeilen nach oben automatisch versetzt, um damit im unteren Teil des Bildschirms für weitere „Druck“-Anweisungen Platz zu schaffen. Das bedeutet, daß der gesamte Bildschirminhalt jeweils um eine Zeile nach oben versetzt wird. Der Quirleffekt ist nichts anderes als eine optische Täuschung, die durch die Anzahl der Zeichen verursacht wird, die der VC-20 auf den Bildschirm setzt.

Um das Programm anzuhalten, drücken Sie die RUN STOP -Taste.



Jetzt sind Sie wieder an der Reihe. Geben Sie die folgenden zwei Zeilen ein:

1 SPACE P R I N T " H A L L O IHR NAME " ;

und RETURN betätigen.

R U N und RETURN -Taste betätigen.

Wie Sie leicht sehen können, ist es ohne Probleme möglich, zum Fernseh-Star zu werden. Das Ausmaß der optischen Täuschung im Zusammenhang mit der Bewegung des allgemeinen Bildes hängt von der Anzahl der sich in der Nachricht befindenden Zeichen ab.

Auch hier wiederum kann die Verarbeitung angehalten werden, indem die RUN STOP -Taste gedrückt wird.

Das erste graphische Zeichen

Zunächst einmal müssen Sie den Bildschirm löschen, indem Sie die Umschalttaste SHIFT und CLR HOME -Taste drücken. Danach sollten Sie SHIFT und S drücken. Auf dem Bildschirm erscheint nun ein blaues *Herz*. Damit haben Sie ein erstes *graphisches Zeichen* eingegeben.

Versuchen Sie doch nun einmal, andere graphische Zeichen einzugeben. Halten Sie die Taste C= und drücken Sie einige graphische Zeichen. Dabei handelt es sich um den linken vorderen Teil der Beschriftung der jeweiligen Taste. Diese Zeichen sind besonders dazu geeignet, Formulare, Diagramme und Graphiken zu gestalten. Die gleichzeitige Betätigung der Umschalttaste SHIFT und der Taste C= ermöglicht die Eingabe von Groß- und Kleinbuchstaben. Siehe Kapitel 3 für weitere Erklärungen.

Die Bildschirmgröße und -Kapazität

Wie groß ist der Bildschirm des VC-20? Wollen Sie dies herausfinden? Löschen Sie jetzt den Bildschirm und schreiben Sie folgendes:

N E W und Drücken der RETURN -Taste

1 P R I N T " ♥ " ; RETURN -Taste



Daraufhin füllt sich der Bildschirm mit blauen Herzen. Wenn Sie jetzt die Herzen je Zeile auszählen werden Sie feststellen, daß sich in jeder Zeile 22 davon befinden. Das bedeutet, daß Ihnen der VC-20 22 „Druck"-Positionen (PRINT) zur *Verfügung stellt.* Manchmal redet man auch von *Spalten* anstatt Positionen. Der VC-20 hat also 22 Spalten.



Schauen Sie nun, wieviel *Zeilen* Sie finden. Wird die CTRL -Taste gedrückt, dann blinken die vier letzten Zeilen in langsamer Folge. Sie erkennen nun leicht: der VC-20 verfügt über 23 Zeilen.

SPALTEN

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

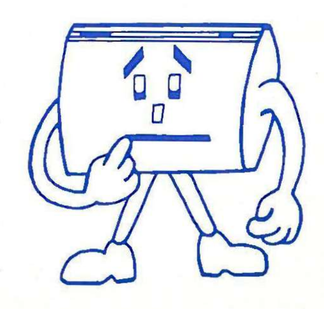
ZEILEN

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22

22 Spalten

23 Zeilen

Nach
Adam Riese:
23 x 22 =
506
Positionen

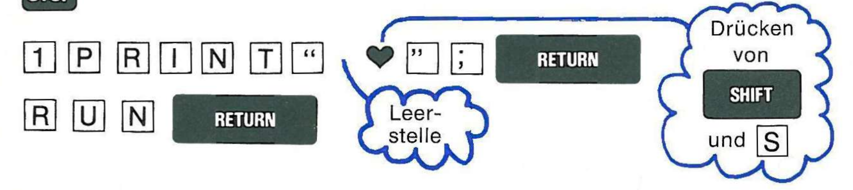


Das bedeutet, daß der VC-20 über *506* Bildschirm-Positionen verfügt, die mit Zeichen, Buchstaben, Symbolen usw. ausgefüllt werden können. Der VC-20 ist in der Lage, mit bis zu 506 Zeichen gleichzeitig zu jonglieren. Erstaunlich, nicht!

VC-20 TIP:

Der VC-20 verfügt also über 22 Spalten in jeder Zeile. Wenn die Länge einer Druckanweisung ein glatter Teiler von 22 ist, (Nachrichten mit 2 Zeichen, 11 oder 22 Zeichen), dann druckt (PRINT) der VC-20 in übersichtlicher Spaltenform. Jede andere Nachrichtenlänge wird zur Ursache haben, daß die entsprechende Nachricht in die nächste Zeile überläuft. Überprüfen Sie einmal diese Behauptung.

Halten Sie den „Druck"-Vorgang (PRINT) der Herzen an, indem Sie die RUN STOP -Taste drücken. Geben Sie jetzt die folgenden Zeilen ein:



Es ist möglich, die Art und Weise des „Drucks" auf dem Bildschirm zu ändern, indem Leerstellen zwischen die gesetzten Anführungszeichen gesetzt werden. Eine andere Möglichkeit besteht in der Hinzufügung von Punkten anstatt der Leerstellen. Sie sollten jetzt versuchen, Ihren Namen und 3 Punkte in das Programm, am Anfang des Kapitels, einzugeben.

Eine Reise über die VC-20-Tastatur

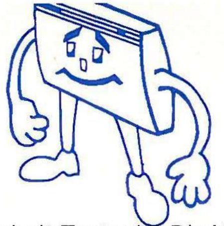


Bis jetzt haben Sie sich damit befaßt, die Tastatur einzusetzen um Programme zu erstellen und zu „Drucken", graphische Zeichen auf den Bildschirm zu bringen, die Verarbeitungsgeschwindigkeit zu steuern (CTRL und RUN STOP) und möglicherweise die ursprünglich eingegebenen Informationen zu korrigieren und zu modifizieren (INST DEL).

Nehmen Sie sich jetzt einmal die Zeit, sich die Tastatur näher zu betrachten um herauszufinden, was Sie mit ihr alles anfangen können. Anhand obiger Grafik werden Sie leicht erkennen, wie umfangreich und leistungsfähig die Tasten angeordnet sind.

Hier haben wir eine Rücksetzungs-Taste („RESET"): Halten Sie die RUN/STOP-Taste und drücken Sie gleichzeitig RESTORE . Damit haben Sie den Computer in den Einschaltzustand zurückgesetzt (so, als hätten Sie ihn eben eingeschaltet). Allerdings bleiben im Speicher die Programme erhalten und können natürlich gelistet (LIST) oder gestartet (RUN) werden.

Dafür können Sie die folgenden Tasten benutzen.



SHIFT

SHIFT-Tasten (Umschalt-Tasten) – Die VC-20-Tastatur ist ähnlich der einer normalen Schreibmaschine und verfügt über 2 *Umschalttasten* und über eine SHIFT-LOCK-Taste (um die Tastatur permanent in den Shift-Modus zu versetzen). Die SHIFT-Taste wird in Verbindung mit anderen Tasten benutzt, um graphische Zeichen einzugeben und Funktionen, wie z. B. die Löschung des Bildschirminhalts auszuführen.



Durch die Betätigung der **CLR-HOME-Taste** wird der Cursor in den linken oberen Teil des Bildschirms positioniert (die Ausgangsposition). Wird die SHIFT-Taste und gleichzeitig die CLR-HOME-Taste gedrückt, dann wird zusätzlich der Bildschirminhalt gelöscht.





Durch die Betätigung der **CRSR-Tasten** ist es möglich, den Cursor in alle gewünschten Richtungen auf dem Bildschirm zu bewegen. Die CRSR-Tasten sind mit einer *Wiederholungs-Einrichtung* ausgestattet, welche den Cursor fortlaufend in die gewünschte Richtung bringt, wenn sie ständig gedrückt wird. Jede Taste ist mit Pfeilen beschriftet, welche die entsprechende Richtung anzeigen – hoch/runter und rechts/links. Um den Cursor nach links oder nach oben zu versetzen, ist es notwendig, zur gleichen Zeit auch die Umschalttaste (SHIFT) zu betätigen. Beachten Sie, daß der Cursor an jede beliebige Stelle des Bildschirms bewegt werden kann, *ohne das sich evtl. an dieser Stelle befindliche Zeichen zu löschen.*

RETURN

Am Ende einer jeden Anweisungszeile drücken Sie die **RETURN-Taste**. Dadurch wird entweder die erstellte Programmzeile gespeichert oder die Anweisung (im Direktmodus) ausgeführt. Es ist vielleicht eher behilflich, sich die RETURN-Taste als ENTER-Taste (Dateneingabe-Taste) vorzustellen, da die Betätigung dieser Taste bewirkt, daß die entsprechenden Informationen gespeichert werden.



Die **CTRL-Taste** benutzen Sie um *Farben* für die Bildschirmdarstellung auszuwählen. Sie können sie aber auch benutzen um eigene *Steuerbefehle* zu definieren. Einige Einschubmodule bedienen sich der CTRL-Taste, um spezielle Funktionen aufzurufen. Diese Taste arbeitet genauso wie die SHIFT-Taste; Sie müssen sie also festhalten, während Sie eine Farbtaste drücken.



Farbtasten – Es ist möglich, die Farbe der auf dem Bildschirm angezeigten Zeichen zu bestimmen, indem gleichzeitig die Tasten CTRL und

die jeweilige Farbtaste gedrückt werden. Die zur Verfügung stehenden Farbtasten (Nummerntasten) befinden sich in der obersten Reihe der Tastatur und sind entsprechend der Farben beschriftet. Die 8 Farbtasten sind wie folgt: BLK (black = schwarz), WHT (white = weiß), RED (= rot), CYN (cyan = türkis), PUR (purple = violet), GRN (green = grün), BLU (blue = blau) und YEL (yellow = gelb). Mittels dieser Tasten ist es möglich, die gewünschten Farben für Buchstaben, Nummern und graphische Zeichen auszuwählen oder zu ändern, egal, ob dies innerhalb oder außerhalb eines Programms geschehen soll. Wenn eine Farbe ausgewählt wurde, werden alle darauf folgenden Eingaben in der jeweiligen Farbe auf dem Bildschirm erscheinen, so lange, bis eine Änderung erfolgt.



Mit den Tasten **RVS/ON und RVS/OFF** kann in Verbindung mit der CTRL-Taste die Darstellungsart der einzelnen Zeichen bestimmt werden. Das bedeutet, daß im Falle der Betätigung von CTRL und RVS/ON die Zeichen in negativer Darstellung auf den Bildschirm gebracht werden. Z. B. kann der VC-20 angewiesen werden, weiße Zeichen auf blauem Hintergrund darzustellen (ganz im Gegensatz zur normalen Darstellungsweise). Die normale Bildschirm-Darstellungsweise tritt wieder in Kraft, wenn nochmals die Tasten CTRL und RVS/OFF gleichzeitig gedrückt werden. Sie sollten einige Experimente damit machen.



Die Betätigung der **RUN/STOP-Taste** weist den VC-20 an, daß er mit der augenblicklich durchgeführten Verarbeitung *anzuhalten* hat. Das bedeutet z. B., daß die *Programmausführung* durch die Betätigung dieser Taste angehalten wird. Die gleichzeitige Betätigung der Umschalttaste (SHIFT) und der RUN/STOP-Taste veranlaßt den VC-20, Informationen von einem möglicherweise angeschlossenem Cassettenrecorder in den Speicher zu laden.



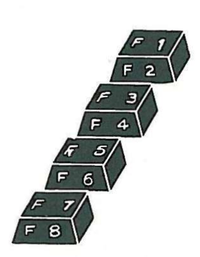
Durch die Betätigung der **INST/DEL-Taste** ist es Ihnen möglich, entweder *neue* Zeichen der angesteuerten Zeile *hinzuzufügen* oder Zeichen von dieser Zeile zu löschen. Wird nur die Taste gedrückt (DELETE = Löschung), wird das sich unmittelbar links vom Cursor befindende Zeichen *gelöscht*. Befindet sich der Cursor in der Mitte einer Zeile, dann wird das linksbündige Zeichen gelöscht und alle zur rechten Seite des Cursors stehenden Zeichen werden um eine Stelle nach links versetzt. Die gleichzeitige Betätigung der Umschalttaste (SHIFT) und der INST/DEL-Taste bedeuten für den VC-20, daß neue Zeichen dieser Zeile hinzuzufügen sind. Der VC-20 reagiert darauf, indem an der Stelle des Cursors etwas Platz geschaffen wird, so daß die neuen Zeichen *hinzugefügt* werden können. Dies ist eine sehr wichtige Funktion, die hauptsächlich für die Weiterbearbeitung und Korrektur von ursprünglich eingegebenen Fehlern eingesetzt werden kann.

Graphische Zeichen / Commodore Taste
Beim Einschalten des Geräts befindet sich dieses automatisch im Graphik-Modus, was bedeutet, daß GROSSBUCHSTABEN und mehr als 60 auf der Tastatur vorhandene graphische Zeichen eingegeben werden können. Auf jeder Taste sind zwei graphische Zeichen. Für die sich auf der rechten Seite befindenden

graphischen Zeichen ist die SHIFT-Taste und die entsprechende andere Taste zu betätigen. Für die links liegenden graphischen Zeichen ist die „COMMODORE"-Taste und die jeweilige andere Taste zu drücken. Auf diese Weise ist es möglich, sowohl die GROSSBUCHSTABEN als auch die volle Auswahl der graphischen Zeichen einzugeben. Es bleibt Ihrer Phantasie überlassen, durch die Eingabe entsprechender Kombinationen von Zeichen (nebeneinander oder auch übereinander) Bilder, Tabellen und Graphiken zu entwickeln.



Groß- und Kleinbuchstaben und Graphik-Tasten – Werden die Tasten SHIFT und COMMODORE *zur gleichen Zeit* gedrückt, dann wird der VC-20 in den *Textmodus* versetzt. Damit ist es möglich, die Tastatur des Systems wie die einer gewöhnlichen Schreibmaschine einzusetzen, wozu noch die auf der linken Seite der Tasten befindlichen angesiedelten graphischen Zeichen hinzukommen. Diese eignen sich besonders gut für die Erstellung von Diagrammen, Graphiken und Formularen. Die Zurückschaltung in den Modus „Großbuchstaben / vollständiger graphischer Zeichenvorrat" wird wiederum durch das gleichzeitige Drücken von SHIFT und COMMODORE bewerkstelligt.



Programmfunktions-Tasten – Die sich auf der rechten Seite der Tastatur befindenden vier Tasten haben zum Zeitpunkt des Einschaltens des Geräts keine Funktion. Ihnen können jedoch Aufgaben / *Funktionen* übertragen werden, ausgehend von Anwendungen, die von Ihnen entwickelt werden können. Durch das gleichzeitige Drücken oder Nicht-Drücken der SHIFT-Taste ergibt sich eine Summe von 8 Tasten, denen Funktionen oder Aufgaben zugewiesen werden können. Diese Programmfunktions-Tasten werden häufig in Verbindung mit Einschubmodulen verwendet, die Sonderprogramme enthalten. Natürlich ist auch ein Programmierer in der Lage, diesen Tasten besondere Funktionen und Aufgaben zuzuweisen.

Sondertasten – Die VC-20-Tastatur bietet Sonderzeichen, die bei den meisten Schreibmaschinen und sogar bei den meisten Computern nicht zur Verfügung stehen. Dazu gehören z. B. pi (π), ←, ↑, > sowie < [und].

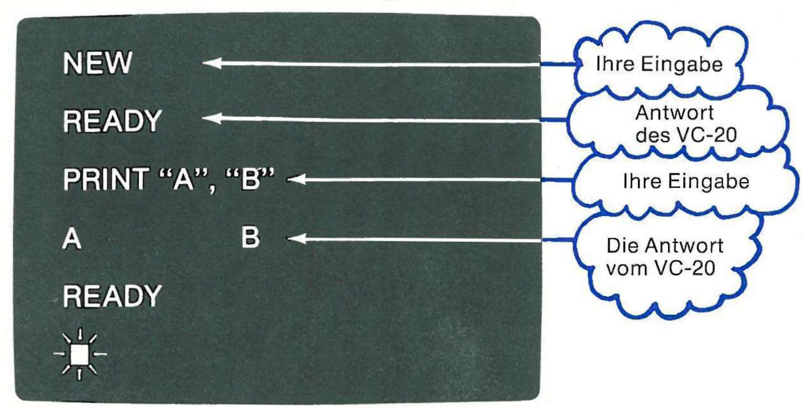
Damit ist die Reise über die VC-20-Tastatur beendet. Es ist jedoch äußerst schwierig auch nur annähernd einen Überblick über die Einsatzmöglichkeiten dieser äußerst leistungsfähigen Tastatur zu geben. Daher ist es wärmstens zu empfehlen, daß Sie sich einen eigenen „Fahrplan" machen. Sie sollten nach Herzenslust die Tastatur erforschen und auf Herz und Nieren prüfen. Sie sollten Ihrer Experimentierfreudigkeit keinen Zwang antun und die Mannigfaltigkeit der einzelnen

Einrichtungen sowohl im umgeschalteten Modus (SHIFT) als auch im nicht umgeschalteten Modus austesten. Dazu sollten Sie die Vielfalt von Möglichkeiten graphischer Zeichen einsetzen. Die Tastatur ist die direkte Verbindung zum VC-20. Die Beherrschung der Tastatur bedeutet gleichzeitig auch die Beherrschung des VC-20.

Nach Beendigung der Tests ist der Bildschirm zu löschen und es sind folgende Daten einzugeben:

NEW RETURN PRINT „A", „B" RETURN

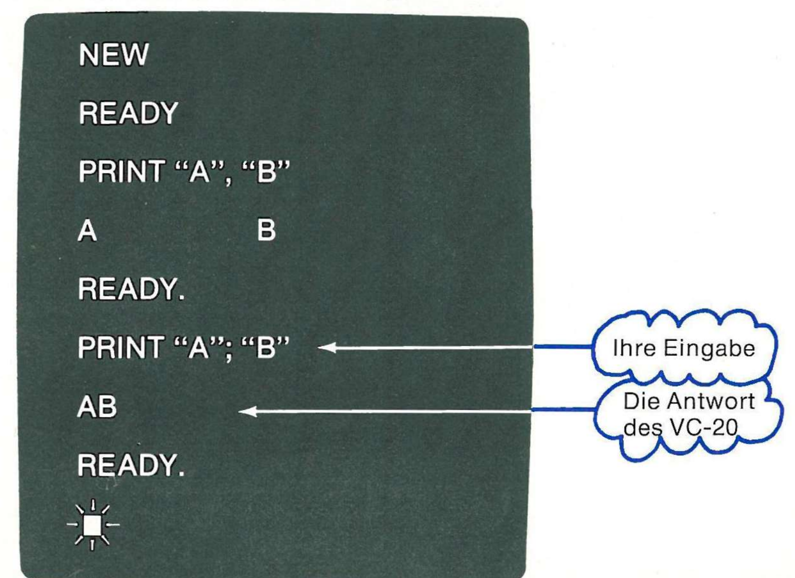
Auf dem Bildschirm erscheint folgendes:



Geben Sie jetzt folgendes ein:

PRINT „A"; „B" RETURN -Taste

Der Bildschirminhalt sieht wie folgt aus:



Bei der Eingabe der ersten PRINT-Anweisung wurde zwischen die zwei Buchstaben „A“ und „B“ ein Komma gesetzt. Der VC-20 nahm dies zum Anlaß, mehrere Leerstellen zwischen die zwei Buchstaben zu bringen. Bei der zweiten Eingabe der PRINT-Anweisung wurde ein Strichpunkt zwischen die zwei Buchstaben gesetzt, was für den VC-20 bedeutete, daß die zwei Buchstaben direkt nebeneinander zu setzen sind.

Im ersten Fall ist das „B“ 11 Stellen von dem Buchstaben „A“ entfernt. Daran erkennen Sie, daß der VC-20 den Bildschirmbereich in zwei gleichgroße Teile aufteilt.

Wenn der VC-20 zwei Nachrichten auf den Bildschirm zu „drucken“ (PRINT) hat, die durch ein *Komma* getrennt sind, dann wird die erste Nachricht auf die linke Seite und die zweite Nachricht auf die rechte Seite des Bildschirms gebracht . . .

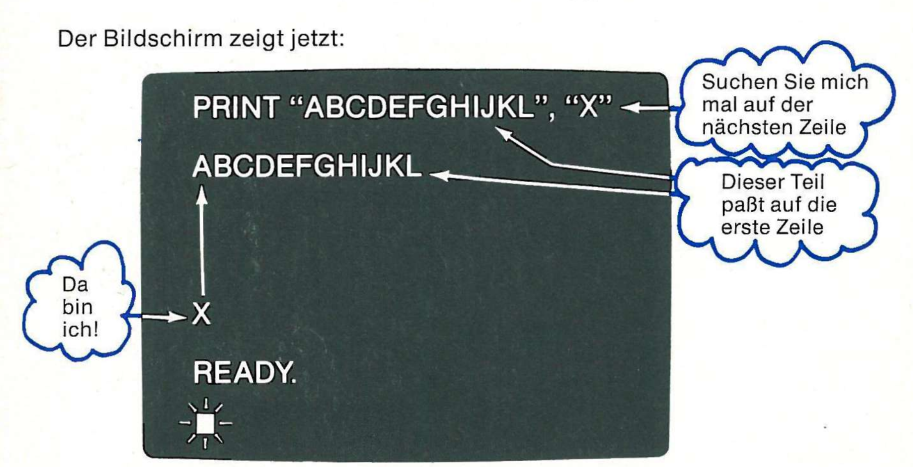
es sei denn

daß der *erste* Teil mehr als 11 Zeichen beinhaltet.

Wenn die erste Nachricht gleich oder kleiner als 11 Zeichen ist, „druckt“ der VC-20 diese Information und springt zur Mitte der jeweiligen Bildschirmzeile, um die zweite Nachricht auszugeben. Ist die Nachricht größer als 11 Zeichen, dann erscheint der zweite Teil der Information auf der nächsten Zeile. Löschen Sie den Bildschirm und probieren Sie folgendes Beispiel:

PRINT „ABCDEFGHIJKL“, „X“

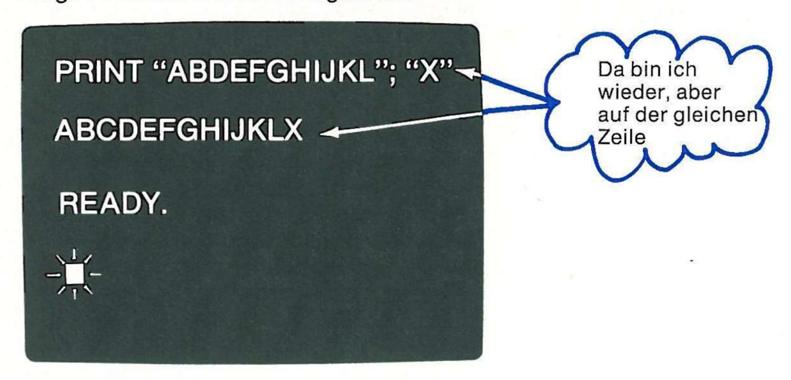
Der Bildschirm zeigt jetzt:



Der erste Teil der Nachricht ist 12 Zeichen lang, was bedeutet, daß das „X“ auf der nächsten Zeile landet. Jetzt sollte dieses Beispiel unter Verwendung des Strichpunktes (;) ausprobiert werden.

PRINT „ABCDEFGHIJKL“; „X“

Zeigt Ihr Bildschirm dies Ergebnis?

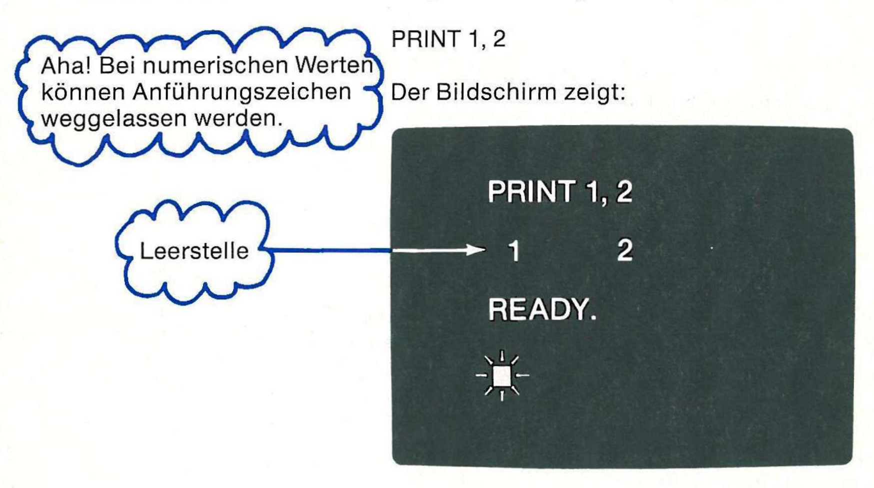


Haben Sie das verstanden? Der VC-20 benimmt sich wie eine Schreibmaschine mit einem automatischem *Tabulator,* der auf die Mitte der Bildschirmzeile zielt. Wird ein Komma registriert, springt der VC-20 entweder zum zweiten Teil der Bildschirmzeile (Mitte der Bildschirmzeile) oder zum Anfang der nächsten Zeile, abhängig davon, welche Position als nächste zur Verfügung steht.

Löschen Sie den Bildschirm und geben Sie folgende Zeile ein:

PRINT 1, 2

Der Bildschirm zeigt:

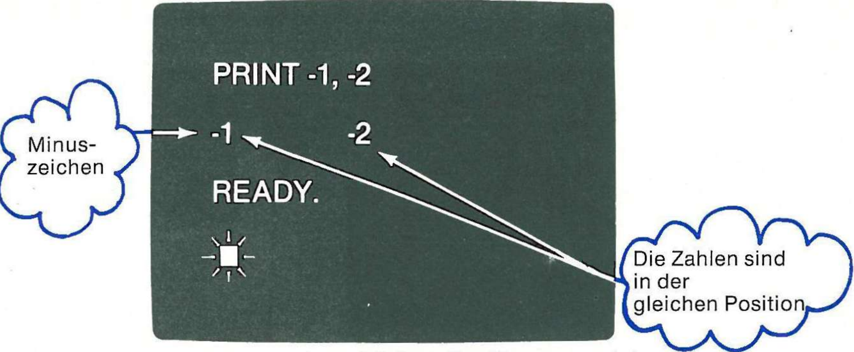


Sie haben sicherlich bemerkt, daß sich vor der ersten Nummer eine Leerstelle befindet. Da die Position mit einem *arithmetischen Vorzeichen* belegt werden kann, wird sie absichtlich von dem VC-20 frei gehalten. Ist die Zahl positiv, dann bleibt die Position leer. Wenn die Zahl negativ ist, dann wird *ein Minuszeichen* (-) eingetragen.

Sie sollten auch das ausprobieren. Dazu sind die folgenden Informationen einzugeben:

PRINT -1, -2

Schauen Sie auf den Bildschirm und beobachten Sie, was passiert:

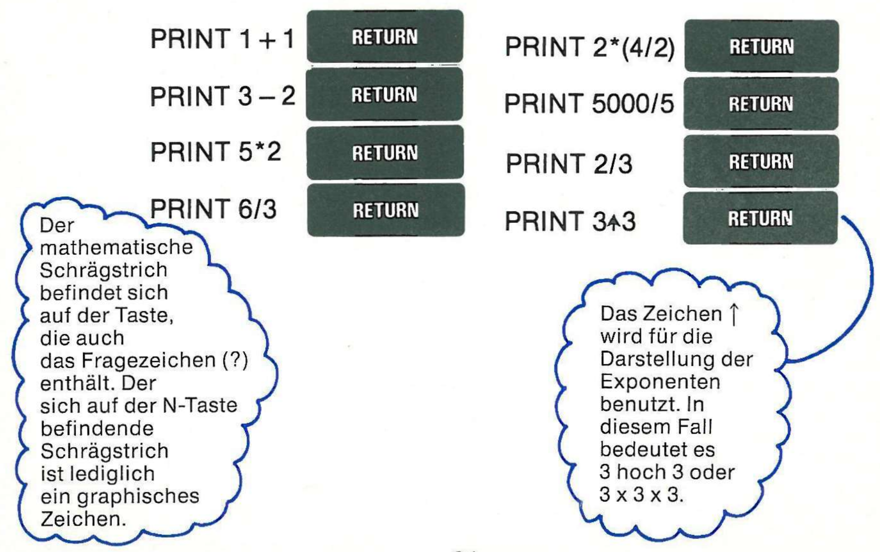


Die Zahlen erscheinen in den gleichen Positionen, welche auch in dem vorhergehenden Beispiel belegt wurden. Jedoch gehen ihnen dieses Mal Minuszeichen voraus.

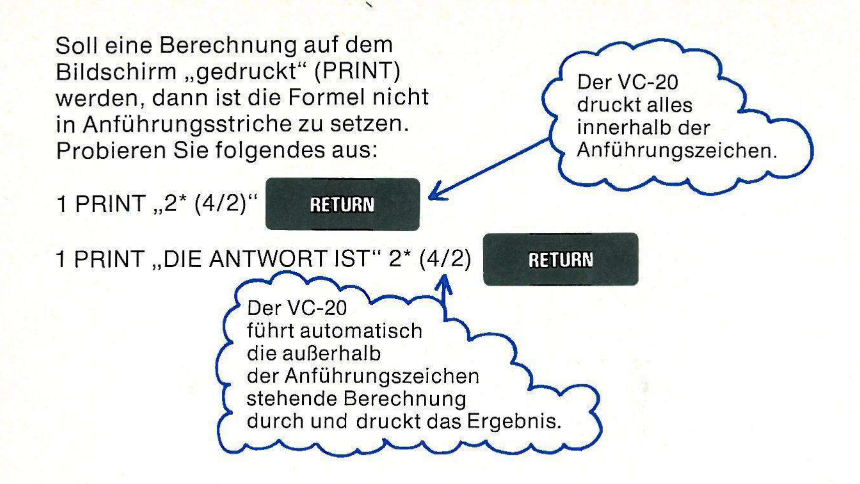
Diese beiden Beispiele sollen Ihnen einen Kurzüberblick darüber geben, wie der VC-20 dabei behilflich ist, Nachrichten und Informationen auf den Bildschirm zu bringen. Es gibt noch viele andere, dieser Funktion dienlichen Einrichtungen des VC-20. Sie werden diese nach und nach während des Ablaufs Ihrer Experimente kennen lernen.

Der VC-20-Rechner

Der VC-20 kann auch als 9stelliger Rechner eingesetzt werden. Die Zeichen + und – wirken sich hier genau so aus, wie auch in der herkömmlichen Vorgehensweise der Lösung mathematischer Aufgaben. Bei dem VC-20 wird die Multiplikation mit einem Sternchen (*) und die Division mit einem Schrägstrich (/) angezeigt. Geben Sie jetzt diese Rechenbeispiele ein und überprüfen Sie die Resultate. Der Anhang C und K gibt weitere Informationen.



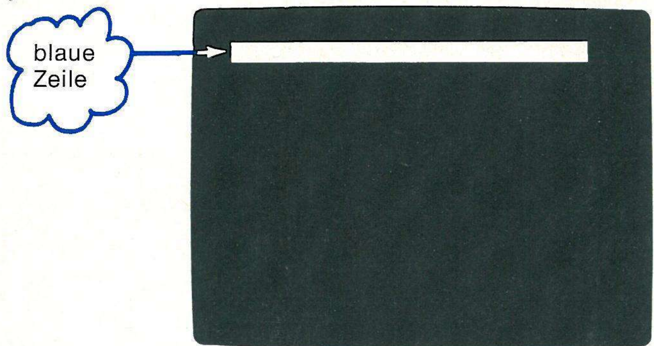
Soll eine Berechnung auf dem Bildschirm „gedruckt“ (PRINT) werden, dann ist die Formel nicht in Anführungsstriche zu setzen. Probieren Sie folgendes aus:



Einführung in die Farbgestaltung

Der VC-20 ist in der Lage, Buchstaben, Nummern und graphische Symbole in acht unterschiedlichen Farben auf dem Bildschirm auszugeben. Darüber hinaus ist es möglich, alle Zeichen in negativer Bildschirmdarstellung erscheinen zu lassen. Nach der Löschung des Bildschirminhalts sollten Sie die CTRL-Taste betätigen und zur gleichen Zeit die folgende Taste: RVS ON

Wenn jetzt die CTRL-Taste losgelassen und die Leertaste ständig gedrückt wird, bildet sich auf dem Bildschirm eine blaue Zeile.



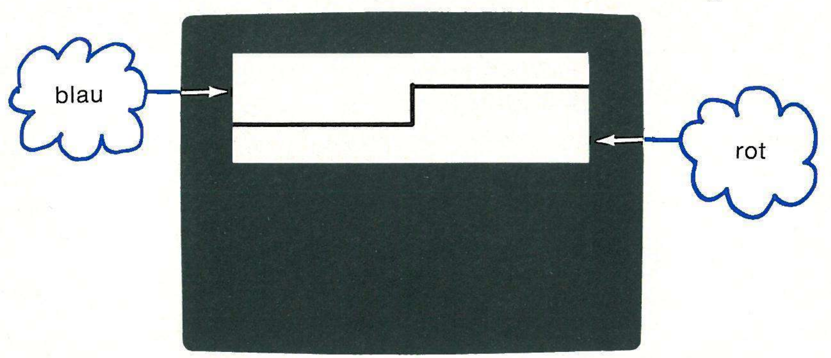
Probieren Sie jetzt, die Leertaste für eine etwas längere Zeit zu drücken. Sobald der Cursor auf der rechten Seite des Bildschirms verschwindet, erscheint er wieder auf der linken Seite in der nächsten Zeile und es bildet sich ein größerer blauer Block.

Lassen Sie nun die Leertaste los und drücken Sie die folgende Tastenkombination:

CTRL und gleichzeitig RED (rot).

Die Nummerntaste 3

Der Cursor sollte jetzt die rote Farbe angenommen haben. Wenn die Leertaste wieder gedrückt wird, bildet sich eine neue rote Zeile. Wenn das der Fall ist, können Sie sich jetzt als „Malermeister" betätigen!



Probieren Sie jetzt nach Herzenslust jede einzelne Farbe, wobei die Bildschirmbereiche so dick oder dünn wie gewünscht ausfallen können. Wirken die Farben des VC-20 nicht aufmunternd?

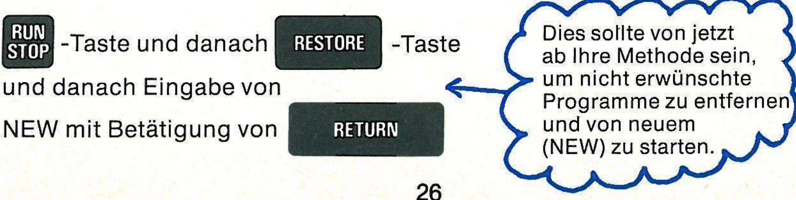
Drücken Sie nun CTRL und Ø RVS OFF danach die Leertaste.

Sie stellen fest, daß eigentlich nichts geschieht, mit der Ausnahme, daß Leerzeichen eingegeben werden. Wird die Tastenkombination CTRL und 9 RVS ON gedrückt, dann erscheinen die Farbzeilen wieder. Jetzt versuchen Sie mal einige Buchstaben in negativer Bildschirmdarstellung einzugeben. Diese Darstellungsart ist hervorragend dazu geeignet, Überschriften sowie besondere Wörter und Zahlen hervorzuheben. Es ist auch möglich, diese Einrichtung innerhalb eines Programms zu aktivieren.

So z.B. ist es möglich, die folgenden Informationen einzuprogrammieren:



Zur Vorbereitung für das nächste Kapitel geben Sie folgendes ein:





Farben und graphische Zeichen

- Programmierung der Farben
- Die Farbtasten des VC-20
- Wechsel der Farben des Bildschirm-Hintergrunds und -Rahmens
- Farbkombinationen für Bildschirm-Hintergrund und -Rahmen
- Farbgestaltung des Bildschirms
- Bildschirmpositionen
- Willkürliche Farbauswahl
- Kombination von Ton und Farben
- Graphische Zeichen der Tastatur
- Graphische Zeichen in Kopfzeilen und Überschriften

Programmvorschlag

Bringen Sie nun den VC-20 mit folgendem Programm zum Laufen!

```
1FOR H = 1 TO 505
2PRINT " ♥ ";
3NEXT
4FORC = 8 TO 255 STEP 17
5POKE 36879, C
6FOR T = 1 TO 500: NEXT
7NEXT
8GOTO 50
RUN
```

Das bedeutet, daß 505 Herzchen auf dem Bildschirm auszugeben sind.

Dieser Befehl ändert den Wert „C“ (Color) mit jedem Schritt um den Wert „17“

Dies ist eine Verzögerungs-Schleife, die erst, nachdem der VC-20 bis „500“ gezählt hat, eine andere Farbe anzeigt.

Eingabe: [R] [U] [N] und Drücken der Taste [RETURN].

Um das Programm in der Ausführung anzuhalten, drücken Sie die [RUN STOP] -Taste.



Farben und graphische Zeichen

Der VC-20 ermöglicht es Ihnen, den gesamten Bildschirm farblich auszufüllen. Als Sie den VC-20 eingeschaltet haben, waren Rahmen, Cursor und eventuell sich auf dem Bildschirm schon befindenden Zeichen in *Farbe* dargestellt. Das ist jedoch nur anfänglich so. Der VC-20 kann 8 Cursor-Farben, 8 Farben für den Rahmen und 16 Hintergrundfarben darstellen.

Sie sollten jetzt versuchen, Farben auf den Bildschirm zu bringen, indem die sich auf der Tastatur des VC-20 befindenden Tasten zur Farbgestaltung eingesetzt werden. Wenn Sie als erstes einen ganz beliebigen Buchstaben eingegeben, sollte dieser in dunkelblauer Farbe auf weißem Hintergrund erscheinen. Jetzt wenden wir uns der oberen Reihe der Tastatur zu, die mit den Nummern 1 bis 8 beschriftet ist. Von vorne gesehen sind die entsprechenden Farben leicht abzulesen. Diese können jetzt in Verbindung mit der CTRL -Taste zum Einsatz kommen. Halten Sie die CTRL -Taste und drücken Sie die YEL -Taste (YELLOW = GELB).

Nach dieser Aktion drücken Sie eine beliebige Taste. Es erscheint das entsprechende Zeichen auf dem Bildschirm in gelber Farbe. Der Farbwechsel kann zu jedem Zeitpunkt geschehen, indem die CTRL -Taste und eine beliebige Farbtaste gleichzeitig gedrückt werden. Die darauf folgende Eingabe von Zeichen erfolgt in der jeweils ausgewählten Farbe.

Halten Sie jetzt die CTRL -Taste und die RVS ON -Taste. Alle darauf folgenden Eingaben von Zeichen erscheinen auf dem Bildschirm in negativer Bildschirmdarstellung, so lange, bis die RETURN -Taste betätigt wird. Diese Darstellungsform ähnelt etwas dem Negativ eines davon abgezogenen *Farbbildes*. Wird jetzt die CTRL -Taste gehalten und die RVS OFF -Taste betätigt, dann erscheinen alle darauf folgenden Eingaben von Zeichen auf dem Bildschirm in normaler Darstellungsform.

Programmierung der Farben

Jetzt ist es an der Zeit, die Farbsteuerung mit einem simplen Programm zu kombinieren. Wenn die CTRL -Taste gehalten und eine Farbtaste innerhalb von Anführungszeichen eingegeben wird, erscheint auf dem Bildschirm ein graphisches Zeichen in negativer Bildschirmdarstellung. Das ist durchaus in Ordnung. Führen Sie jetzt die folgenden Aktionen durch:

Halten der SHIFT -Taste und Betätigung der CLR HOME -Taste.

Jetzt sind folgende Eingaben zu machen:

N E W und Betätigung der RETURN -Taste.

Durch Eingabe von:

1 P R I N T " und SPACE drücken.

Drücken der SHIFT -Taste und Q erzeugt den graphischen Ball.

£ ● SPACE SPACE ← ● SPACE " ;

Dieses Zeichen erscheint, wenn die Tasten CTRL und RED gedrückt werden.

Drücken von CTRL und BLU

und RETURN -Taste

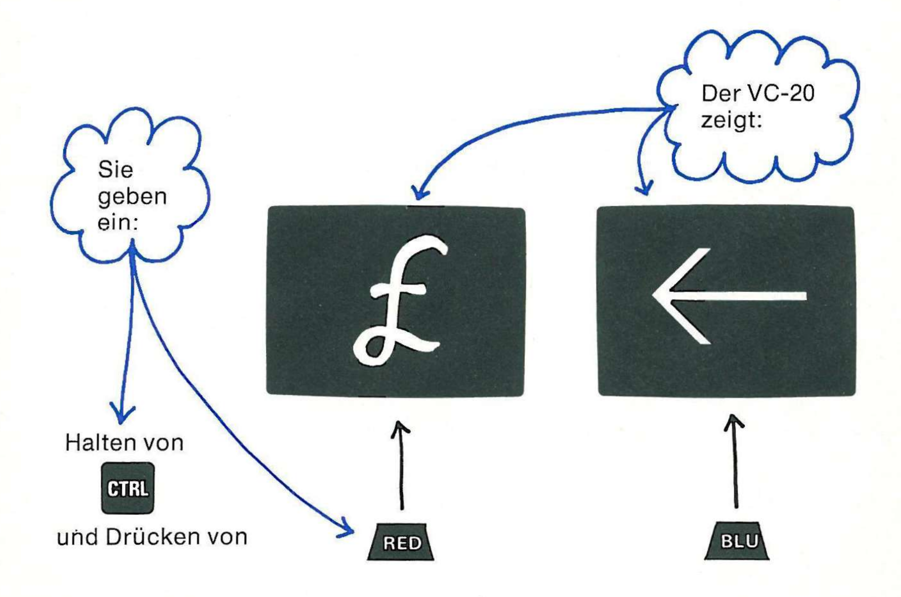
Eingabe von 2 SPACE G O T O SPACE 1

Danach ist RETURN zu drücken.

Wenn Sie Schwierigkeiten haben, dieses Beispiel einzugeben, dann sollten Sie sich nochmals mit dem Abschnitt „Der Anfang" in Kapitel 1 auseinandersetzen. Es gibt keine Eingabenkombination, die dem VC-20 schaden könnte. Es ist jedoch möglich, daß Sie anfänglich durch bestimmte Tastenkombinationen verwirrt werden. Haben Sie so z.B. unabsichtlich die SHIFT LOCK -Taste gedrückt, dann kann es möglich sein, daß die auf dem Bildschirm angezeigten Zeichen sehr schwierig zu entschlüsseln sind. Sollte bei der Eingabe ein Fehler unterlaufen sein, dann empfehlen wir zu diesem Zeitpunkt, die RETURN -Taste mehrere Male zu drücken und die gesamte Zeile neu einzugeben. Die neue Zeile wird die alte Zeile automatisch ersetzen. Sind die zwei oben gezeigten Zeilen auf dem Bildschirm, dann starten Sie mit:

R U N und drücken die RETURN -Taste.

Sobald der letzte RETURN erfolgt ist, sollten hunderte von roten und blauen Bällen über den Bildschirm huschen. Wie ist das möglich? Ganz leicht mit dem VC-20. Wenn man sich obiges Beispiel nochmals betrachtet, dann wird man zwei fremdartige Zeichen auf der mit „1" anfangenden Zeile erkennen. Diese Zeichen wurden dort hingebracht, als die RED -Taste und auch die BLU -Taste in Verbindung mit der CTRL -Taste gedrückt wurden. Diese Symbole bedeuten für den VC-20 in Kurzschrift, daß der erste Ball mit roter Farbe und der zweite Ball mit blauer Farbe auf den Bildschirm zu bringen ist.



Eine Kleinigkeit! Sind Sie der Farben rot und blau überdrüssig, brauchen Sie lediglich die STOP-Taste zu drücken. Es ist wiederum einmal an der Zeit, der eigenen Phantasie freien Lauf zu lassen. Dazu ist die Zeile 1 erneut einzugeben. Jetzt sollten einfach einige Farbtasten in Verbindung mit der CTRL -Taste und einigen graphischen Zeichen sowie Buchstaben und Nummern gedrückt werden. Der VC-20 schafft sie alle und verhilft zu einem kunterbunten Farbenfest.

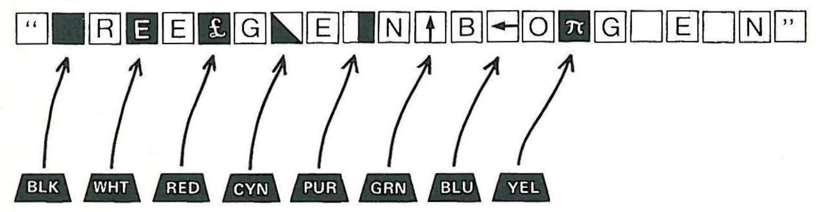
Die Farbtasten des VC-20

In dem letzten Beispiel wurde Ihnen gezeigt, daß die Farbensteuerung in dem PRINT-Befehl eingebettet werden kann, indem in Verbindung mit der CTRL -Taste die folgenden Farbtasten betätigt wurden:

BLK WHT RED CYN PUR GRN BLU YEL

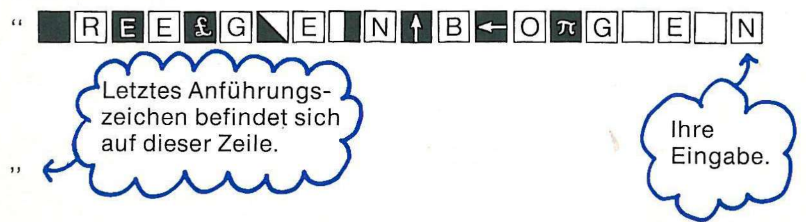
Diese Tasten sind oben mit den Nummern 1 bis 8 versehen. Werden diese Tasten innerhalb von Anführungszeichen gedrückt, dann erscheinen auf dem Bildschirm u. U. sehr fremdartige Zeichen. Um herauszufinden, um welche Zeichen es sich dabei handelt, ist die RUN STOP -Taste zu betätigen (wenn das Programm noch immer läuft). Danach sollte der Bildschirm gelöscht werden und folgende Eingabe durchgeführt werden:

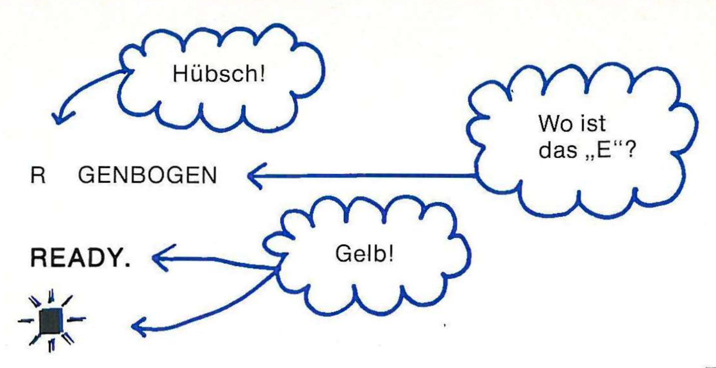
PRINT



Vergessen Sie nicht, die Anführungszeichen zu schreiben oder auch die CTRL -Taste in Verbindung mit der jeweiligen Farbtaste. Der Bildschirm sollte nun so aussehen:

PRINT





Wo ist der Buchstabe „E"? Die zweite ausgewählte Farbe ist WHT (WHITE = WEISS) und der Hintergrund ist . . . – Sie haben es erraten – . . . auch *weiß.* Das führt selbstverständlich zu dem Ergebnis, daß der weiße Buchstabe „E" auf weißem Hintergrund zu einer Leerstelle im REGENBOGEN führt.

Die anderen Buchstaben des Wortes sind andersfarbig. Der letzte Buchstabe, das „N", ist gelb. Dadurch ist auch die READY-Nachricht und der Cursor gelb. *Werden Farbselektionen in eine PRINT-Nachricht eingebaut, dann erinnert sich der VC-20 an die letzte Farbe und bringt alle Informationen nur noch in dieser Farbe auf den Bildschirm.* Um den Cursor wieder „blau" zu machen (Prosit), ist die Tastenkombination CTRL und BLU (BLUE = BLAU) zu betätigen. Es wird empfohlen, jetzt erst einmal ein wenig mit den verschiedenen Farbkombinationen im Zusammenhang mit der PRINT-Anweisung zu spielen. In den nächsten Zeilen wird beschrieben, wie diese Farbsteuerungen anderweitig eingesetzt werden können. Zuerst jedoch eine sehr wichtige Ankündigung:

ACHTUNG! ACHTUNG!

Die VC-20-Farbenshow wird angekündigt

Wechsel der Farben des Bildschirm-hintergrunds und Rahmens

Zu diesem Zeitpunkt sollten Sie mit dem Wechsel der Farben für Buchstaben und graphische Zeichen vertraut sein. Jetzt ist es an der Zeit zu zeigen, wie die gewünschte Bildschirm-Hintergrundfarbe und die Farbe des Bildschirmrahmens ausgewählt werden kann. Es stehen 255 Farbkombinationen zur Verfügung. Um den VC-20 dafür betriebsbereit zu machen, drücken Sie erst die RUN STOP -Taste und danach die Tastenkombination SHIFT und CLR HOME, um den Bildschirminhalt zu löschen. Schreiben Sie danach folgende Zeile:

N E W und RETURN -Taste

(Zum Ende einer jeden der folgenden Eingabezeilen ist die RETURN -Taste zu drücken.)

Das am linken Rand stehende erste Zeichen einer jeden Zeile.

POKE was?

```
1FOR X = 1 TO 255
2POKE 36879, X
3PRINT "♥ POKE 36879," X
4FOR T = 1 TO 1000: NEXT T
5NEXT X
```

Dafür ist die Tastenkombination SHIFT und CLR HOME zu betätigen.

Addiere zur variablen „X" den Wert „1" – mit anderen Worten – Änderung von „X" auf den nächsten Wert und erneute Ausführung.

Dies ist eine Zeitverzögerungs-Schleife, die den VC-20 veranlaßt, von 1 bis 1000 (SPITZE!) zu zählen.

Nach der Eingabe dieser Zeilen sollten Sie diese nochmals auf ihre Richtigkeit kontrollieren, um sicherzustellen, daß diese im Einklang mit den hier aufgeführten Informationen stehen. Werden Differenzen festgestellt, empfehlen wir, diese Zeilen nochmals von Anfang an neu einzugeben. Sie können zur entsprechenden Zeile gehen, indem die Taste CRSR verwendet wird. Fällt die Überprüfung der Eingaben positiv aus, ist es zu empfehlen, Schreibblock und Bleistift parat zu halten. Nun starten Sie mit:

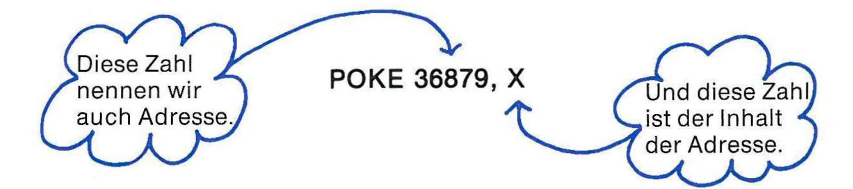
R U N und RETURN .

Der Bildschirm sollte jetzt in allen erdenklichen Farbkombinationen erstrahlen. Sowohl der Bildschirmrahmen als auch der eigentliche Bildschirm-Hintergrund wechseln seine Farben. Sogar die Zeichen im oberen Teil des Bildschirms unterliegen diesem Farbwechsel. Es werden 255 unterschiedliche Farbkombinationen angezeigt. (Sollte eine der Zeilen einen Syntax-Fehler anzeigen, schlagen wir vor, diese Zeile nochmal einzugeben und

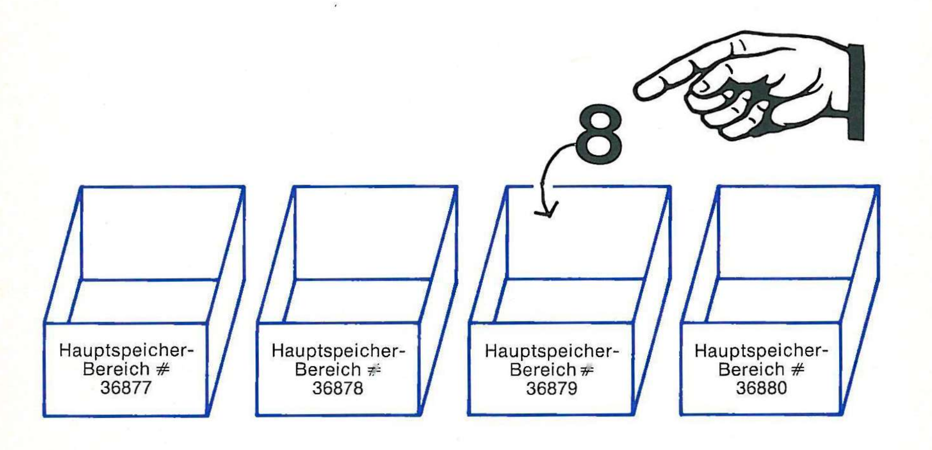
R U N und RETURN abermals zu drücken.)

Wenn Sie während dieses Farbspiels einer bestimmten Farbkombination besonders angetan sind, empfehlen wir, die Nummer dieser Kombination aufzuschreiben. Nur die sich am Ende der Nachricht befindende Nummer ändert sich. Der Befehl „POKE 36879" bleibt wie gehabt. (Ab und zu passiert es, daß die Nachricht nicht gelesen werden kann, weil die Farbe der Nachricht die gleiche ist, wie die des Hintergrunds. Das sind die Tücken der Technik. Die Buchstaben und Zahlen werden nach einigen anderen Farbkombinationen wieder erscheinen.) Der VC-20 ist mit 8 Farben für den Bildschirmrahmen, 16 Farben für den Bildschirm-Hintergrund und 8 Farben für die eigentlichen Zeichen ausgestattet. Es ist möglich, alle im Zusammenhang mit den Zeichen kombinierbare Farben für jeden Hintergrund auszuwählen. Die gebotenen Farbkombinationen sind also enorm. Soll der Bildschirm zur ursprünglichen Farbe zurückkehren, dann drücken Sie RUN STOP und RESTORE .

Die Zeile 2 des obigen Beispiels ist für den Farbwechsel verantwortlich. Diese Zeile enthält einen POKE-Befehl. Jeder dieser Befehle besteht aus zwei numerischen Werten, die vom VC-20 verwendet werden:



Die erste Zahl (in diesem Fall 36879) ist eine Stelle im Hauptspeicher (man kann dabei an ein kleines Kästchen denken, welches mit der Adresse 36879 versehen ist), in welche die zweite Zahl (X) zu plazieren (POKE) ist. Die Hauptspeicherstelle 36879 ist die, in welcher der VC-20 die Information über die Farben des Bildschirmrahmens und Bildschirm-Hintergrunds aufbewahrt. Jeder Wert „X" entspricht einer anderen Farbkombination auf dem Bildschirm. „X", fängt mit einem Wert von 1 an, wechselt über zu 2, 3, usw., bis zu einem entgültigen Wert von 255.



BEISPIEL = POKE 36879, 8

Um Ihnen bei der Suche nach der *gewünschten* Farbkombination behilflich zu sein, haben wir hier eine Tabelle der POKE-Werte (X) aufgestellt, welche die Farben des Bildschirm-Hintergrunds und des Bildschirmrahmens anzeigt. Alle möglichen Kombinationen sind hier vertreten. Die POKE-Werte sind in aufsteigender Reihenfolge aufgeführt, wobei immer Sprünge von 8 X-Werten gemacht werden. Diese hier fehlenden POKE-Werte verursachen die Anzeige der Zeichen in negativer Bildschirmdarstellung.

Farbkombinationen für Bildschirm-Hintergrund und -Rahmen

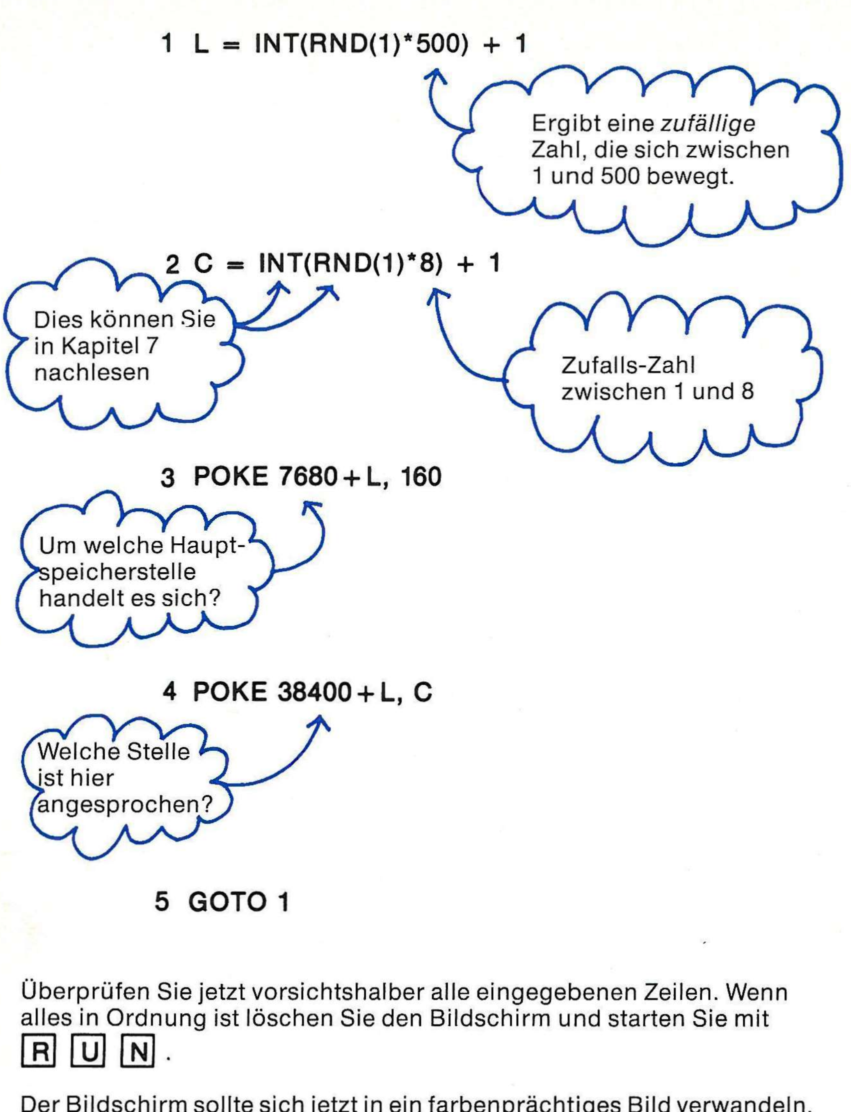
Hintergrund	Bildschirmrahmen BLK SCHW	WHT WEISS	RED ROT	CYAN TÜRKIS	PUR VIOLETT	GRN GRÜN	BLU BLAU	YEL GELB
SCHWARZ	8	9	10	11	12	13	14	15
WEISS	24	25	26	27	28	29	30	31
ROT	40	41	42	43	44	45	46	47
TÜRKIS	56	57	58	59	60	61	62	63
VIOLETT	72	73	74	75	76	77	78	79
GRÜN	88	89	90	91	92	93	94	95
BLAU	104	105	106	107	108	109	110	111
GELB	120	121	122	123	124	125	126	127
ORANGE	136	137	138	139	140	141	142	143
HELLORAN.	152	153	154	155	156	157	158	159
ROSA	168	169	170	171	172	173	174	175
HELLTÜRKIS	184	185	186	187	188	189	190	191
HELLVIOL.	200	201	202	203	204	205	206	207
HELLGRÜN	216	217	218	219	220	221	222	223
HELLBLAU	232	233	234	235	236	237	238	239
HELLGELB	248	249	250	251	252	253	254	255

Farbgestaltung des Bildschirms

Es ist jetzt an der Zeit, die graphischen Einrichtungen und die unterschiedlichen Farbkombinationen des VC-20 nach Lust und Laune einzusetzen. Auch in diesem Zusammenhang brauchen nur wenige Zeilen eingegeben zu werden, den VC-20 zu veranlassen, mit der farbenprächtigen Show zu beginnen.

Befindet sich ein Programm noch in der Ausführung, dann kann der VC-20 lediglich durch die Betätigung der [RUN STOP] -Taste angehalten werden. Löschen Sie den Bildschirm und geben Sie die folgenden geheimnisvollen Zeilen ein.

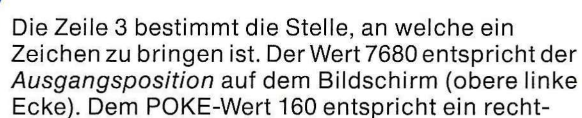
[N] [E] [W] (bis jetzt noch nicht sehr geheimnisvoll).



Überprüfen Sie jetzt vorsichtshalber alle eingegebenen Zeilen. Wenn alles in Ordnung ist löschen Sie den Bildschirm und starten Sie mit [R] [U] [N].

Der Bildschirm sollte sich jetzt in ein farbenprächtiges Bild verwandeln. Mit nur wenigen Befehlen entstand ein farbenfrohes Feuerwerk. Wie wurde das bewerkstelligt?

Ganz einfach ausgedrückt wählen die Zeilen 1 und 2 zufällig aus, an welche Positionen des Bildschirms (L = Position) welche Farben (C = Farbe) zu kommen haben, wobei die Farbselektion ganz willkürlich erfolgt. Die Zeile 1 generiert irgendeine Nummer, die zwischen 1 und 500 liegt. Die Zeile 2 generiert eine Nummer zwischen 1 und 8.



Die Zeile 3 bestimmt die Stelle, an welche ein Zeichen zu bringen ist. Der Wert 7680 entspricht der *Ausgangsposition* auf dem Bildschirm (obere linke Ecke). Dem POKE-Wert 160 entspricht ein rechteckiges Kästchen (im gewissen Sinne ein Leerzeichen im negativen Bildschirm-Darstellungsformat).

Die Zeile 4 bringt die Farbe auf das Zeichen. Der Wert 38400 entspricht der Hauptspeicher-Stelle der ausgewählten Farbkomponenten des Zeichens in der *Ausgangsstellung* (links oben).

Bildschirmpositionen

VC-20 TIP:

Um Zeichen auf den Bildschirm zu bringen (POKE), muß die entsprechende Bildschirmposition mit dem POKE-Befehl und der jeweiligen Farbe für jedes Zeichen angesteuert werden. Die Hauptspeicherstelle für die Ausgangsposition (links oben) des Bildschirms ist 7680. Die Hauptspeicherstellen für die Farben fangen bei 38400 an. Siehe auch Anhang für die Bildschirmaufteilung.

Versuchen Sie jetzt mal mit anderen Werten auf den Bildschirm zu POKEn. Dazu ist der Wert 160, der sich in der 3. Zeile befindet, auf einen Wert zu ändern, der zwischen 0 und 255 liegt.

Experimentieren Sie – und lassen Sie sich von dem großen Farbspektrum des VC-20 unterhalten.

Der Bildschirm verfügt über 506 Bildschirmpositionen. Daher ist es möglich, Wörter, Buchstaben, Sätze, graphische Zeichen, usw. an jede beliebige Stelle des Bildschirms zu bringen (POKE). Stellen Sie sich einfach vor, daß der Bildschirm aus 506 kleinen Kästchen besteht, wobei jedes mit einer Adresse versehen ist.

Wenn Sie sich mit den POKE-Befehlen „sattgePOKEt" haben, kann es weiter im Text gehen. Wenden wir uns nochmal den Farbsteuertasten zu.

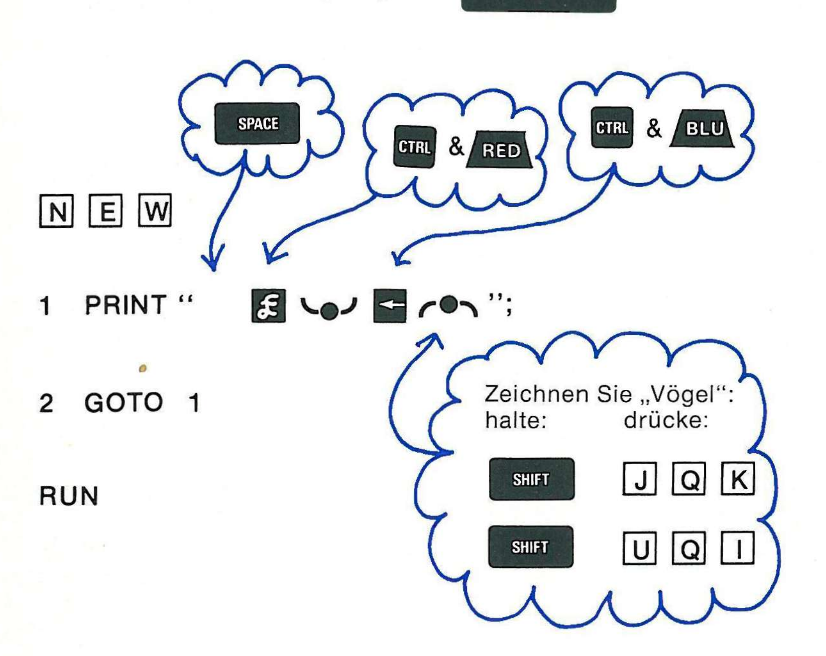
VC-20-TIP:

Soll der Bildschirm (Hintergrund und Rahmen) wieder seine ursprüngliche Farbe haben, dann geben Sie lediglich den folgenden POKE-Befehl ein:

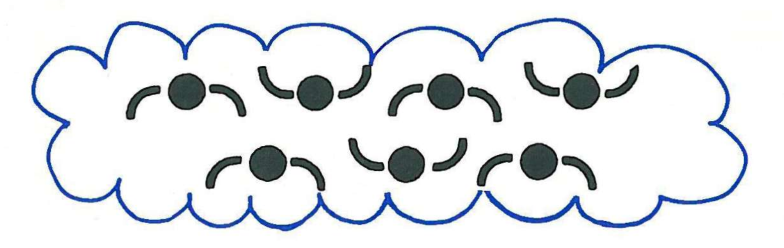
POKE 36879, 27 und RETURN

Zufällige Farbauswahl

Lassen wir doch mal den VC-20 Farben für den Bildschirm selbst auswählen. Dazu löschen Sie zuerst den Bildschirm und geben dann die folgenden Zeilen (immer gefolgt von RETURN ein:



Rote und blaue Vögel überall. Wohin fliegen sie? Sie bewegen sich wirklich!



Da auch Vögel einmal müde werden, verhelfen Sie ihnen zu einer Ruhepause, indem Sie die Taste RUN STOP drücken. Nun, da Sie wissen, wie man „Vögel" (auf dem Bildschirm) zeichnet, löschen Sie diesen und schreiben Sie folgende Zeilen: (nicht vergessen, die RETURN -Taste zu betätigen).

N E W

1 A$ = " [Farbsteuerzeichen] "

Für die Eingabe dieser Zeile halten Sie die CTRL -Taste und drücken Sie jede der COLOR-Tasten.
(BLK WHT RED CYN PUR GRN BLU YEL).

2 N = INT(RND(1)*8) + 1

Diese Zeile erzeugt *zufällige* Zahlen zwischen 1 und 8.

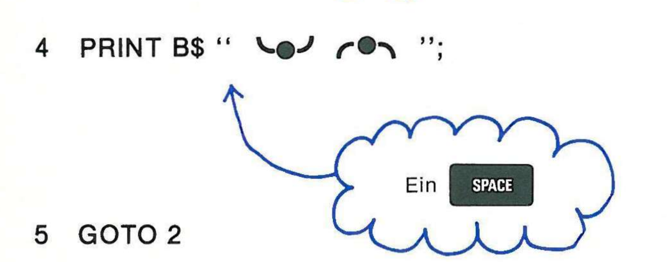
3 B$ = MID$(A$,N,1)

Farbsteuerung

Zufälliger Wert

Wählt ein Farbsteuerzeichen aus A$

4 PRINT B$ " [Vogelzeichen] ";



5 GOTO 2

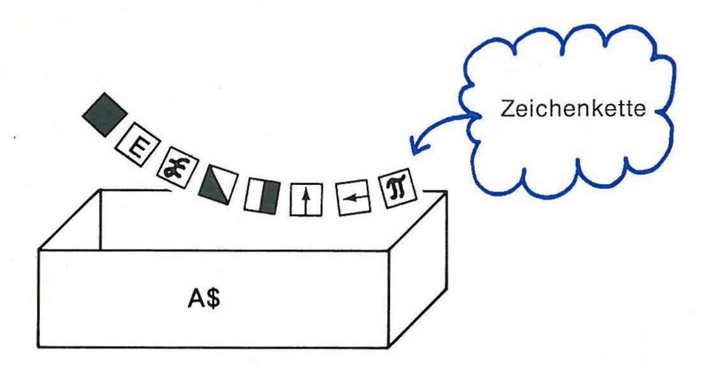
Wir raten Ihnen, die Eingaben nochmals zu überprüfen. Löschen Sie nun den Bildschirm und starten Sie mit [R] [U] [N] . Im Falle einer Fehlermeldung raten wir Ihnen, die fehlerhafte Zeile nochmals neu einzugeben um erneut mit [R] [U] [N] zu starten. Vögel! Vögel! und noch mehr Vögel! Wo man auch hinsieht sind Vögel mit unterschiedlichen Farben!! Ab und zu treten auch Lücken im „Vogel"-Schwarm auf. An diesen Stellen bringt der VC-20 einen weißen Vogel auf weißem Hintergrund. Sollen diese unsichtbaren „Vögel" auch in Erscheinung treten, dann muß die Hintergrundfarbe geändert werden. (Vergessen wie? Dazu ist lediglich eine Zahl in die Adresse 36879 zu POKEN.)

Haben Sie aber eine andere Hintergrundfarbe gewählt, z. B. blau, dann verschwinden selbstverständlich alle blauen Vögel. Die verdiente Ruhepause wird den Vögeln durch das Drücken der RUN STOP -Taste gewährt.

Das letzte Beispiel verwendete einige sehr interessante Möglichkeiten des VC-20, womit die Vögel auf den Bildschirm gelangten und dazu veranlaßt wurden, ihre Farben zu wechseln. Führen Sie sich deshalb einmal jede Zeile einzeln zu Gemüte. Die erste Zeile ist:



Alle Farbsteuerzeichen werden zu einer *Zeichenkette* (STRING) zusammengesetzt und in der variablen A$ (sprich „A-String") gespeichert. Stellen Sie sich dabei A$ als ein Kästchen vor, welches in der Lage ist, alpha-numerische Zeichen aufzunehmen. Die erste Position in dem Kästchen A$ enthält das Farbsteuerzeichen für BLK (SCHWARZ). Die letzte Position enthält das Farbsteuerzeichen YEL (GELB).



Es folgt die Zeile:

2 N = INT(RND(1)*8) + 1

Diese Zeile erzeugt eine *zufällige* positive Zahl zwischen 1 und 8, und speichert sie in der Variablen „N". „N" ist eine Variablen-Speicher-Stelle, die in der Lage ist, numerische Werte aufzunehmen. Weil sich hinter dem „N" kein Dollarzeichen ($) befindet, weiß der VC-20, daß es sich hierbei nur um numerische Werte handeln kann.

Die in Zeile 1 zu speichernden Farb-Steuerzeichen wurden in A$ gespeichert. Durch das Dollarzeichen erkennt der VC-20, daß die in dieser Speicherbezeichnung stehenden Werte alpha-numerischer Natur, also Zeichenketten sein müssen.

Um an weitere Informationen für die Befehle [I] [N] [T] und [R] [N] [D] (für die Erzeugung zufälliger Zahlen) zu gelangen, sollte Kapitel 7 dieses Handbuches eingehend studiert werden. Für diesen Teil reicht es aus zu wissen, daß diese Zeile Zahlen erstellt, die zwischen 1 und 8 liegen. Die sich rechts des Gleichheitszeichens (=) befindende Zahl ist der Maximalwert, bis zu dem eine willkürliche Zahl erzeugt werden darf. Eine Änderung auf z. B. 6 würde bewirken, daß nur Zahlen zwischen 1 und 6 erzeugt werden.

Lassen Sie uns zur nächsten Zeile gehen:

3 B$ = MID$(A$,N,1)

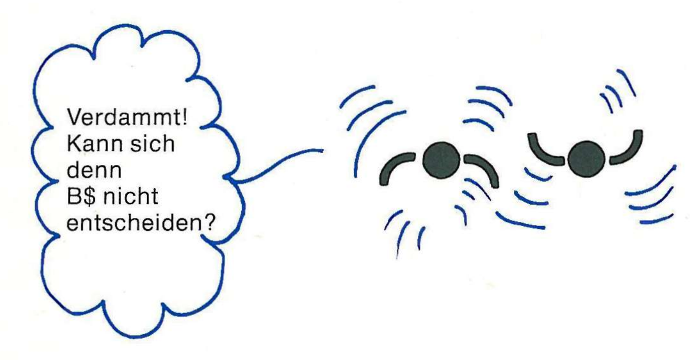
Diese Zeile spricht eine neue Variable mit dem Namen B$ (sprich: „B-String") an und bringt in diese nur ein Farb-Steuerzeichen (aus A$). Die Variable A$ enthält alle 8 Farb-Steuerzeichen. „N" bezieht sich auf eine Nummer von 1 bis 8. Wofür ist MID$ da? Dieser Befehl pickt ein Farb-Steuerzeichen aus der Kette A$ heraus und zwar aus der „N"-ten Position. Dieses Zeichen wird in die Variable B$ kopiert. Zusammengefaßt bedeutet das, daß dieses Programm mit drei kleinen Kästchen (Variablen) oder Hauptspeicher-Stellen arbeitet – A$, „N" und B$. Jedes Mal, wenn der VC-20 auf Zeile 2 trifft, wird ein neuer Wert für „N" vergeben. Dieser Befehl entscheidet, welche Farbe von MID$ aufgelesen und in B$ hineingestellt wird.



In der vierten Zeile steht:

```
4 PRINT B$ "  ";
```

Diese Zeile weist den VC-20 an, das in B$ (sprich: „B-String") stehende Farb-Steuerzeichen für die Zeichnung der „Vögel" zu verwenden. Der Befehl PRINT B$ hat die gleiche Wirkung, als wäre hier eine ganz bestimmte Farbtaste gedrückt worden. Durch die Benutzung von B$ ist es jedoch möglich, den VC-20 dazu zu veranlassen, den Farbwechsel automatisch vorzunehmen. In dem Moment, in dem der Wert in „N" wechselt (Zeile 2), ändert sich auch der Inhalt von B$ (Zeile 3).



Kombination von Ton und Farbe

Die Fähigkeit des VC-20 zufällige Zahlen zu erzeugen, kann mit Farben, Tönen und Graphik-Einrichtungen in einer Vielfalt von interessanten Möglichkeiten kombiniert werden. An dieser Stelle begeben wir uns wieder in das zufällige Farbenspektakel und erzeugen – wenn wir schon einmal dabei sind – auch einige Töne.



Mit diesem letzten Beispiel wird von dem VC-20 eine prächtige Kombination von Farben und Tönen auf willkürliche Art erzeugt. Geben Sie dazu folgende Zeilen ein:

```
NEW

1POKE 36878, 3

2C = INT(RND(1)*255) + 1

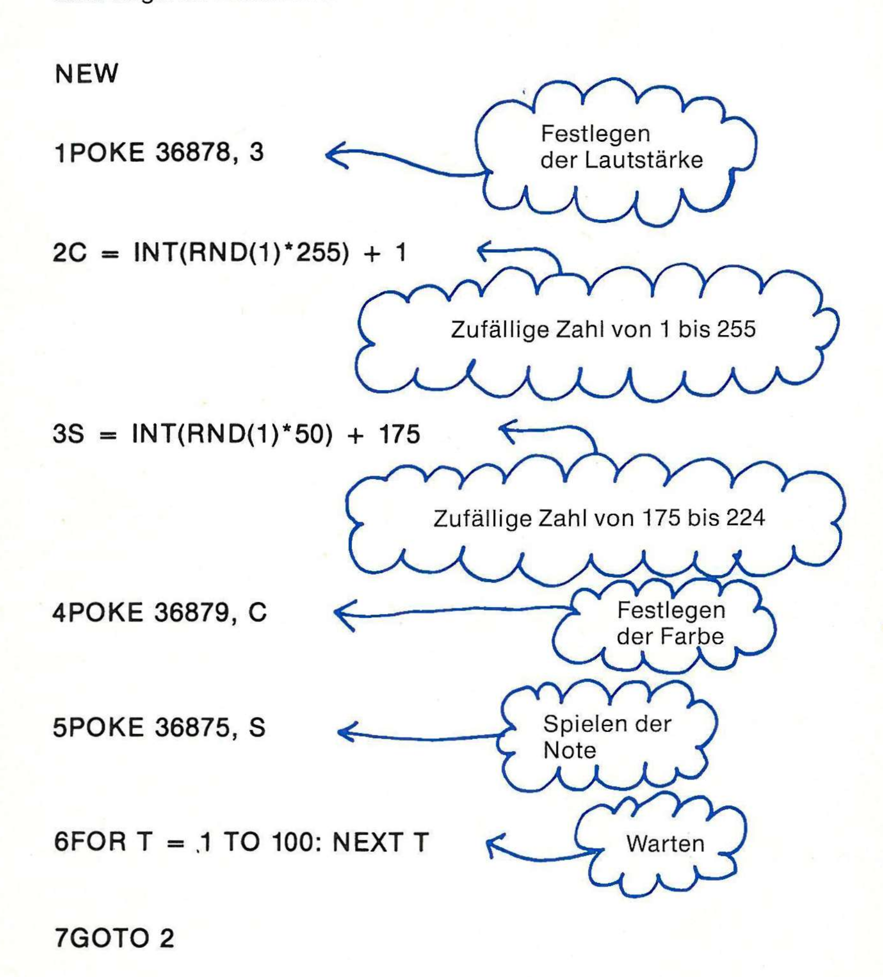
3S = INT(RND(1)*50) + 175

4POKE 36879, C

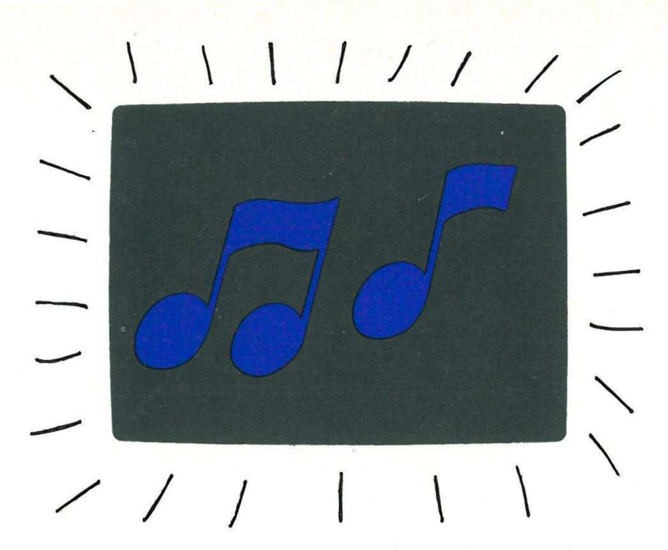
5POKE 36875, S

6FOR T = 1 TO 100: NEXT T

7GOTO 2
```

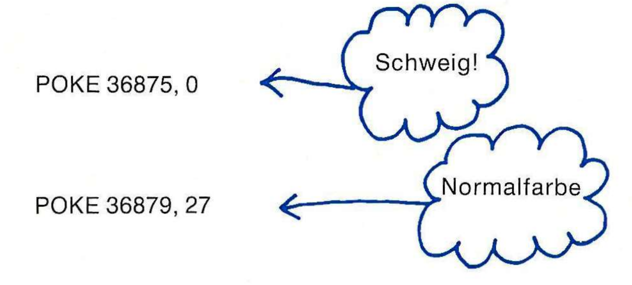


Die Erklärungs-Wölkchen erzählen eigentlich schon die ganze Geschichte. Mehr über Musik können Sie allerdings in Kapitel 5 erfahren. Wenn Sie wollen, können Sie dieses Beispiel laufen lassen. Löschen Sie den Bildschirm und geben Sie [R] [U] [N] ein. (RETURN)



Ach, welch ein Wirrwarr von Farben und Noten! Aber Sie haben nun, was Sie wollten. Selbstverständlich dürfen wir nicht erwarten, daß Zufalls-Computer-Musik jedem gefällt.

Wenn Sie genug von dieser Farb- und Musikmaschine haben, drücken Sie einfach RUN STOP. Sie müssen jetzt aber noch die nächsten zwei Zeilen eingeben um den VC-20 ganz zum Schweigen zu bringen und dem Bildschirm seine normale Farbe zurückzugeben:



Das gleiche hätten Sie auch durch die Tasten RUN STOP und RESTORE erreicht.

Es ist jetzt wieder an der Zeit, daß Sie Ihre eigenen Experimente probieren. Wir wünschen Ihnen viel Freude an den Farben und Tonfolgen.

Graphische Zeichen der Tastatur

Ein besonderer Vorzug des VC-20 ist in der Vielfalt der graphischen Zeichen zu sehen. Von vorne betrachtet verfügen fast alle Tasten über 2 graphische Zeichen. Es ist möglich, diese Zeichen auf den Bildschirm zu bringen und auch auf einem Matrix-Drucker von COMMODORE, falls dieser zur Verfügung stehen sollte.

Diese graphischen Zeichen lassen sich in Kombination mit der COMMODORE--Taste (Auswahl links) und der SHIFT -Taste (Auswahl rechts) einsetzen. Diese beiden Tasten müssen jeweils gehalten werden, während die entsprechende Taste für das gewünschte graphische Zeichen betätigt wird.

VC-20 TIP:

Vorsicht! Wenn die Tasten SHIFT und COMMODORE gedrückt werden, wird der VC-20 in den Groß-/Kleinbuchstaben-Modus versetzt. In diesem Betriebsmodus können nur die sich auf der linken Seite einer jeden Taste befindenden graphischen Zeichen benutzt werden. Um das System wieder in den Großbuchstaben/Graphik-Modus zurückzuführen, sind die Tasten SHIFT und C= erneut zu drücken.

Die problemloseste Art die graphischen Zeichen des VC-20 einzusetzen besteht darin, die Tasten COMMODORE C= oder SHIFT zu halten, während die jeweils gewünschte Taste für das graphische Symbol gedrückt wird. So z. B. wird durch die Betätigung von:

SHIFT S

ein blaues Herzchen auf dem Bildschirm erscheinen. Es ist jetzt Ihnen selbst überlassen, verschiedene Tasten nach Herzenslust auszuprobieren. Hier aber einige wichtige Graphikzeichen, die Sie möglicherweise gut brauchen können.

GRAPHISCHE ZEICHEN

linksbündig		rechtsbündig	
C=	T	SHIFT	Q
C=	@	SHIFT	W
C=	+	SHIFT	M
C=	B	SHIFT	N

Anmerkung: Hier stehen senkrechte und waagrechte Striche unterschiedlicher Größe zur Verfügung, so daß es möglich ist, Zeichnungen nach Ihrem Wunsch zu entwickeln.

Es ist hierbei zu bemerken, daß die sich auf der linken Seite einer jeden Taste befindenden Zeichen besonders für die Entwicklung von Diagrammen, Graphiken und Formularen geeignet sind. Die sich auf der rechten Seite einer jeden Taste befindenden graphischen Zeichen lassen sich gut für Zeichnungen, Illustrationen und Zeichentrick-Effekte einsetzen . . . sogar für Spielkarten!

Graphische Zeichen in Kopfzeilen und Überschriften

Der Einsatz der graphischen Zeichen ist nicht auf die Zeichentrick-Darstellung und auf einfache Spielereien beschränkt. Es ist möglich, eine ganze Reihe von Sondereffekten einzusetzen, die z. B. die Überschriften in Diagrammen und Graphiken hervorheben und die es Ihnen erlauben, bestimmte Textpassagen eines Programms besonders gut kenntlich zu machen. Die einfachste Art und Weise ein Wort oder eine bestimmte Textpassage von dem übrigen Teil abzuheben wird damit erreicht, daß dieser Teil in negativer Darstellung auf dem Bildschirm erscheint. Dazu sind lediglich die Tasten CTRL und RVS ON zu betätigen, wonach die gewünschte Textpassage eingegeben wird.

Zum Beispiel:

Auf dem Bildschirm erscheint ♥ .

5 P R I N T " SHIFT CLR HOME "

10 P R I N T " CTRL

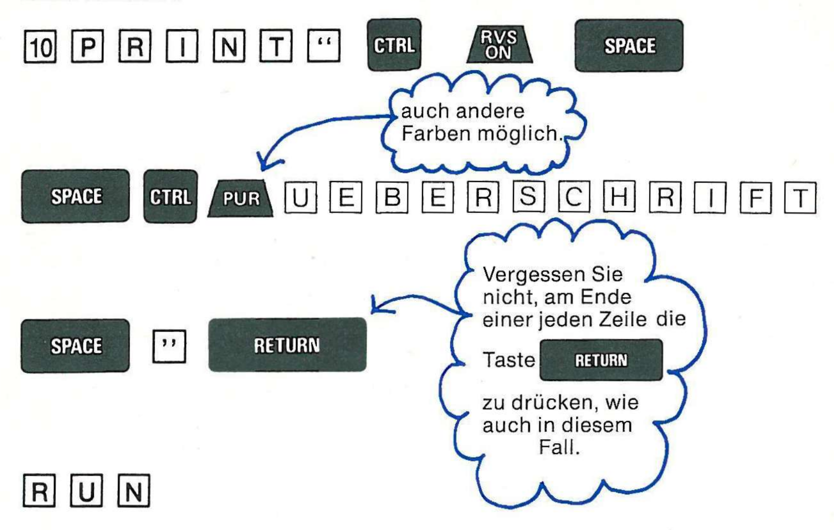
Während der Eingabe halten.

RVS ON U E B E R S C H R I F T "

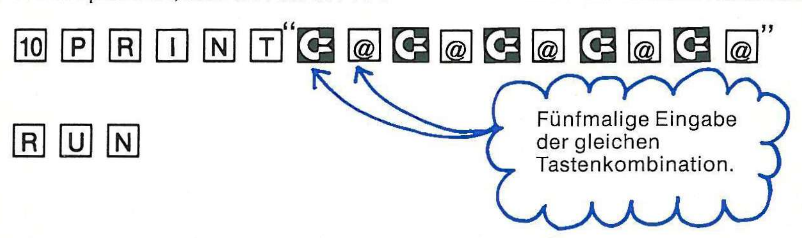
Schreiben einer Überschrift.

R U N

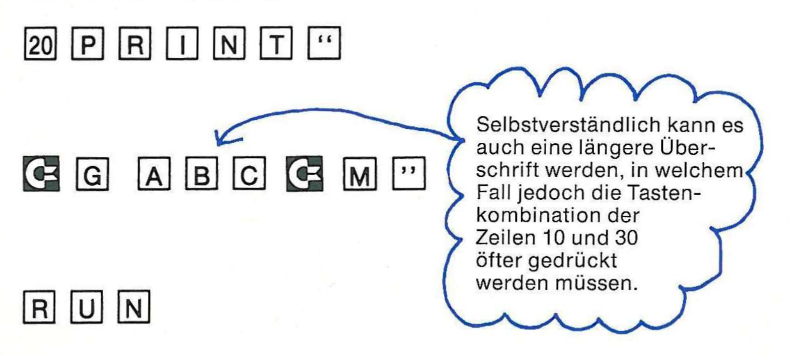
Setzen Sie noch einige Leerzeichen (SPACE), womit eine Überschriftenleiste entsteht.

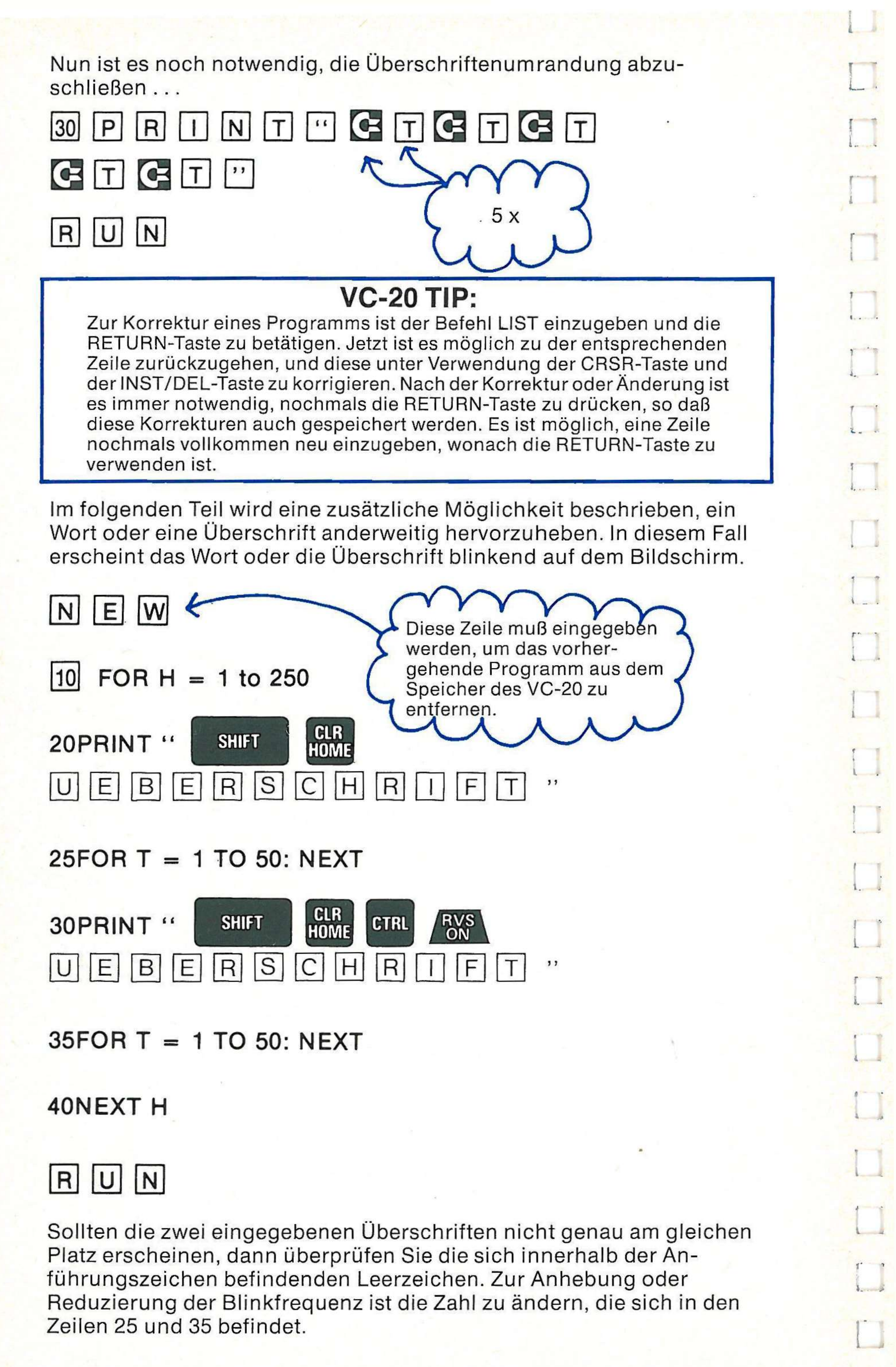


Eine andere Art der Hervorhebung eines Wortes ist die, daß ein Kästchen um das bestimmte Wort gezeichnet wird. Diese Art der Hervorhebung ähnelt sehr stark technischen Zeichnungen. Passen Sie genau auf, daß genau das, was unten aufgezeichnet ist, auch von Ihnen eingesetzt wird. Wir empfehlen, sich mit dieser Technik schrittweise vertraut zu machen.



Auf dem Bildschirm erscheint eine gerade horizontale Linie. Die nächste Zeile erstellt die Überschrift und eine vertikale Linie an dem rechten und linken Ende der Überschrift.





Nun ist es noch notwendig, die Überschriftenumrandung abzuschließen . . .

30 P R I N T " C= T C= T C= T C= T C= T "

R U N

VC-20 TIP:

Zur Korrektur eines Programms ist der Befehl LIST einzugeben und die RETURN-Taste zu betätigen. Jetzt ist es möglich zu der entsprechenden Zeile zurückzugehen, und diese unter Verwendung der CRSR-Taste und der INST/DEL-Taste zu korrigieren. Nach der Korrektur oder Änderung ist es immer notwendig, nochmals die RETURN-Taste zu drücken, so daß diese Korrekturen auch gespeichert werden. Es ist möglich, eine Zeile nochmals vollkommen neu einzugeben, wonach die RETURN-Taste zu verwenden ist.

Im folgenden Teil wird eine zusätzliche Möglichkeit beschrieben, ein Wort oder eine Überschrift anderweitig hervorzuheben. In diesem Fall erscheint das Wort oder die Überschrift blinkend auf dem Bildschirm.

N E W

10 FOR H = 1 to 250

20PRINT " SHIFT CLR HOME
U E B E R S C H R I F T "

25FOR T = 1 TO 50: NEXT

30PRINT " SHIFT CLR HOME CTRL RVS ON
U E B E R S C H R I F T "

35FOR T = 1 TO 50: NEXT

40NEXT H

R U N

Sollten die zwei eingegebenen Überschriften nicht genau am gleichen Platz erscheinen, dann überprüfen Sie die sich innerhalb der Anführungszeichen befindenden Leerzeichen. Zur Anhebung oder Reduzierung der Blinkfrequenz ist die Zahl zu ändern, die sich in den Zeilen 25 und 35 befindet.

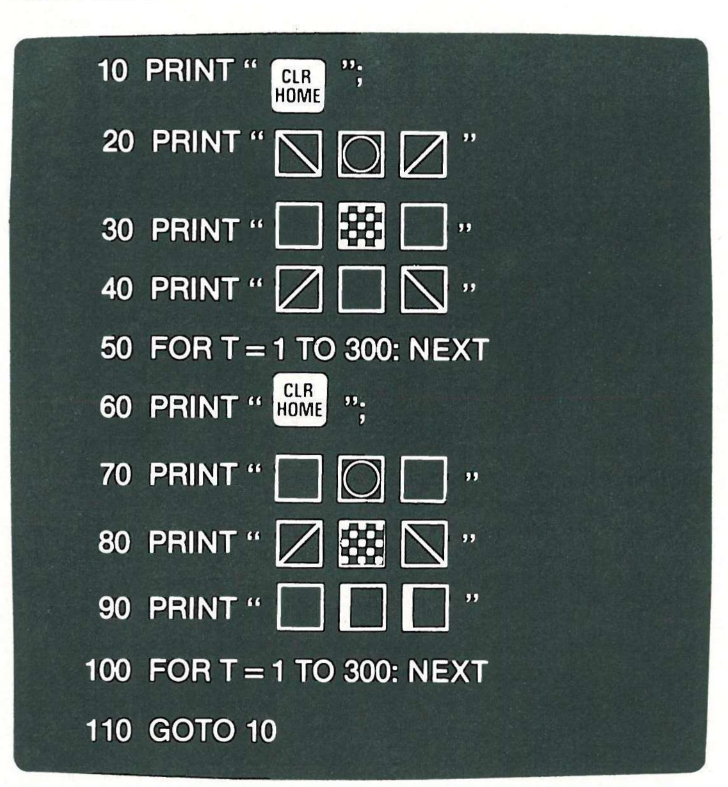


Zeichentrick-Darstellung

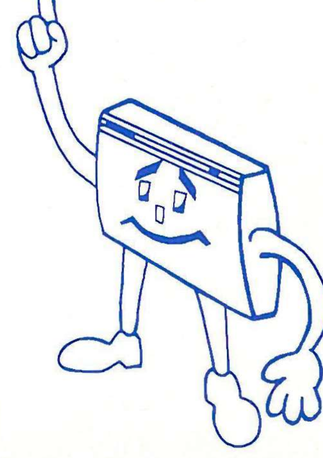
- Fliegende Vögel
- Springender Ball
- Cursor-Steuerung
- Zeichentrick-darstellung mit PEEKS und POKES
- Bildschirm-Hauptspeicher-bereich

Programmvorschlag

Bringen Sie den VC-20 mit folgendem Mini-Programm zum Laufen!



Um das Programm in der Ausführung anzuhalten, drücken Sie die RUN STOP -Taste.

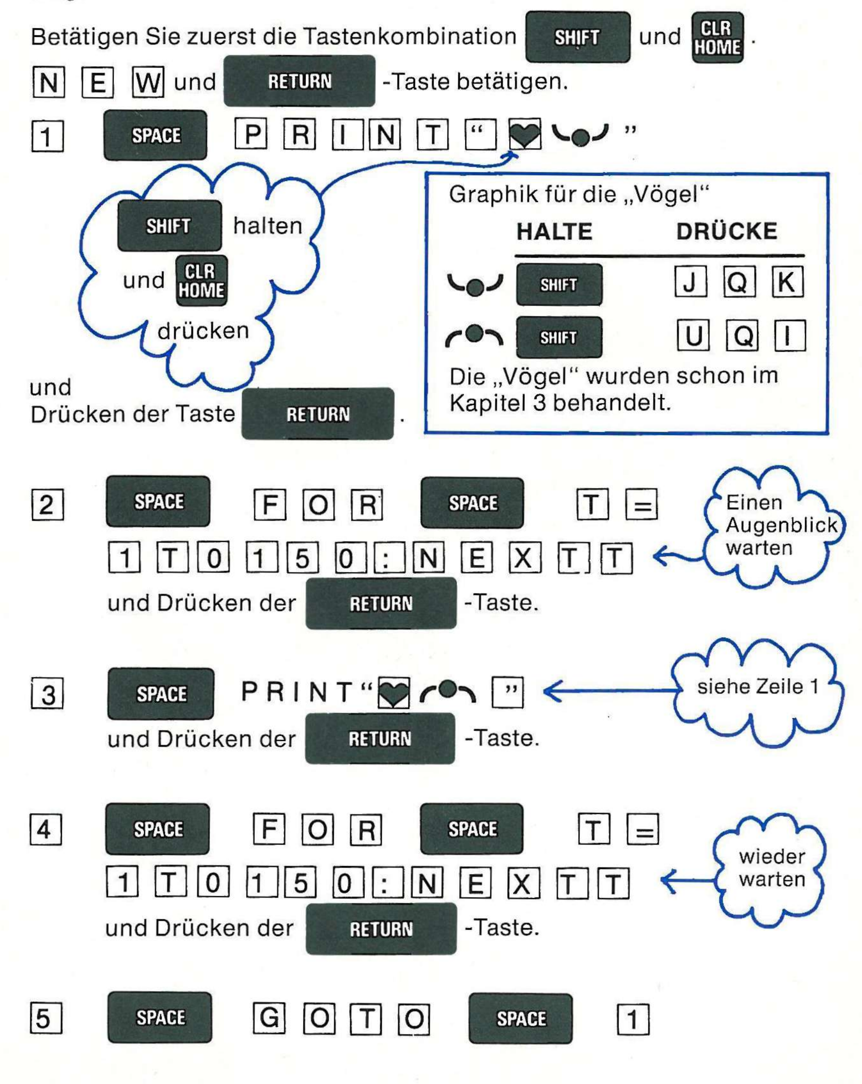


Fliegende Vögel

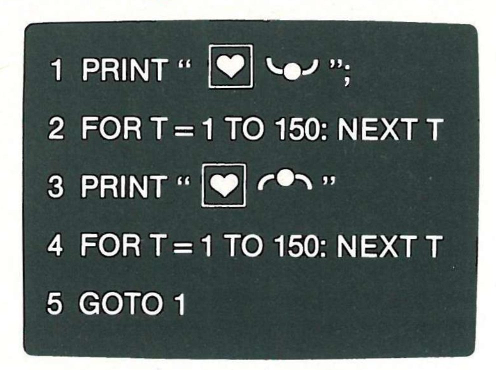
Dieses Kapitel gibt Ihnen eine Einleitung über den Einsatz der graphischen Fähigkeiten des VC-20 zur Erstellung von sich bewegenden Objekten.

Die „Bewegungen" sind eigentlich nichts anderes als eine optische Täuschung, die auch im Zeichentrick-Verfahren angewandt wird. Die Zeichentrick-Darstellung kann jedes Programm interessant machen, egal, ob es sich hierbei um ein Spielprogramm oder ein kommerzielles Programm handelt.

Um mit diesem Verfahren anzufangen, sollten Sie die folgenden Zeilen eingeben:



Die Eingabe sollte also wie folgt aussehen:

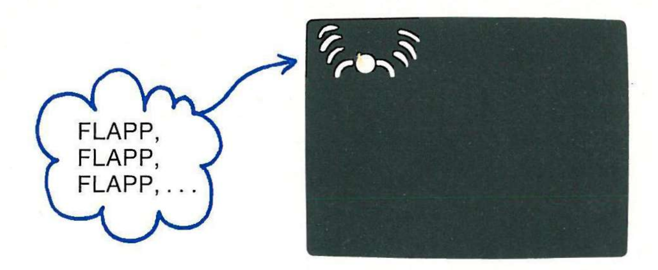


Prüfen Sie das ganze nochmals. Stimmt dies mit der Vorlage überein? Wenn nicht, sind die fehlerhaften Zeilen nochmals einzugeben. Bei Übereinstimmung ist:

R U N und die RETURN -Taste zu drücken.

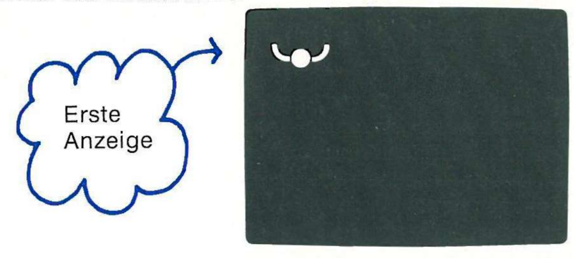
Hinweis: Es ist möglich, eine Zeile neu einzugeben, indem diese überschrieben wird. Die zweite Eingabe wird die erste automatisch entfernen. Es ist möglich, eine Zeile dadurch zu entfernen, indem nur die Zeilennummer eingegeben wird, und danach die RETURN -Taste.

Hier ist Ihre erste Zeichentrick-Darstellung auf dem VC-20. Der in der linken oberen Ecke erscheinende „Vogel" scheint sehr viel Kraft aufwenden zu müssen. Dabei scheint aber ein sehr starker Gegenwind zu herrschen, da sich der Vogel nicht von der Stelle bewegt.



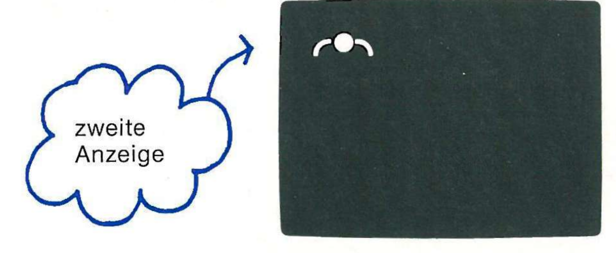
Bei längerer Betrachtung *kann* man sich tatsächlich der Illusion hingeben, daß sich die Flügel bewegen. Das Geheimnis fast aller Zeichentrick-Darstellungen auf dem VC-20 wird in diesem Beispiel gezeigt. Diese optische Täuschung der „Bewegung" läßt sich in einigen wenigen Schritten erklären. Hier ist eine Kurzbeschreibung der Vorgehensweise, wie es möglich ist, daß fast jedes Zeichen winkt, blinkt, sich schüttelt, nickt, sich bewegt, usw.

Erstens: Bringen Sie das Zeichen oder die Zeichenkombination auf eine bestimmte Stelle des Bildschirms.



Zweitens: Verzögern Sie den Programmablauf um eine kurze Zeitspanne. Diese Zeitspanne wird durch die Warte-Schleife FOR-NEXT erreicht. Beispiele dieser Zeitverzögerungs-Schleife werden in den Zeilen 2 und 4 im Programm: „Fliegende Vögel" gezeigt. Diese Zeilen weisen den VC-20 an, von 1-150 zu zählen. Der VC-20 ist in der Lage, sehr schnell bis zu 150 zu zählen. Aber versuchen Sie doch einmal, diese Zahl von 150 auf 500 (oder eine andere Zahl) hochzusetzen und beobachten Sie, was passiert.

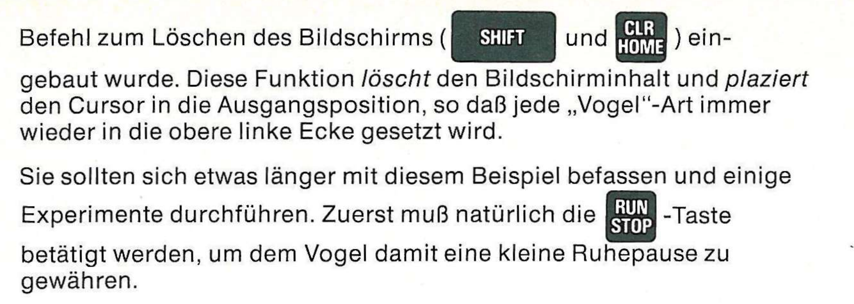
Drittens: Bringen Sie dieses (oder diese) Zeichen in etwas veränderter Form auf die gleiche Bildschirmposition. Durch die vorhergehende Zeitverzögerung kommt die optische Täuschung zustande und das Auge glaubt, eine Bewegung zu sehen.



Viertens: Nun ist es wieder notwendig, den Programmablauf zu verzögern. Ausgehend von dem zu erzielenden Effekt kann diese Warteperiode länger oder kürzer als die vorhergehende sein. Im hier gezeigten Beispiel ist die Zeitverzögerung zwischen den zwei PRINT-Befehlen gleich lang.

Fünftens: Dieser Schritt ist am allerleichtesten, da er lediglich eine *Wiederholung* der ersten vier Schritte bedeutet. Sie erreichen dies, indem Sie den VC-20 durch den GOTO-Befehl anweisen, auf die erste Zeile zurückzugehen und mit der Ausführung von neuem zu beginnen. In diesem Fall bedeutet das nichts anderes, als daß das Programm den „Vogel" (mit Flügeln nach oben) auf den Bildschirm bringt, dann bis 150 zählt, den „Vogel" (mit Flügeln nach unten) auf den Bildschirm in die gleiche Position bringt, danach wieder bis 150 zählt und zurück zum Anfang des Programms geht.

Im ersten und dritten Schritt (wo die entsprechenden Zeichen auf den Bildschirm gebracht werden) ist es oft notwendig sicherzustellen, daß keine überflüssigen Zeichen auf dem Bildschirm sind. In unserem Beispiel wurde das so gelöst, indem in jede „PRINT"-Anweisung der



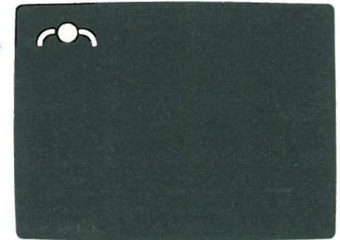
Befehl zum Löschen des Bildschirms (SHIFT und CLR HOME) eingebaut wurde. Diese Funktion *löscht* den Bildschirminhalt und *plaziert* den Cursor in die Ausgangsposition, so daß jede „Vogel"-Art immer wieder in die obere linke Ecke gesetzt wird.

Sie sollten sich etwas länger mit diesem Beispiel befassen und einige Experimente durchführen. Zuerst muß natürlich die RUN STOP -Taste betätigt werden, um dem Vogel damit eine kleine Ruhepause zu gewähren.

Jetzt könnten z. B. die Pausen (150) der Zeilen 2 und 4 geändert werden. Wie wäre es mit der Änderung eines Wertes und der Beibehaltung des anderen Wertes auf 150? Fängt der Vogel an, sich in die Lüfte zu heben? Was passiert, wenn beide Werte auf 50 heruntergesetzt werden? Fliegt der Vogel dann schneller? Wie sieht es mit einem Maximalwert von 500 oder 1000 aus?

Hmmm . . .
der Vogel wirkt
wirklich langsam.





Für den Abschluß ist CLR HOME zu drücken.

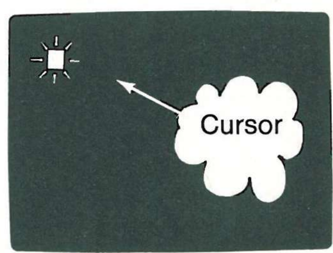
Sie wissen jetzt, daß sich der Bildschirminhalt durch das Halten der SHIFT -Taste und die Betätigung der CLR HOME -Taste löschen läßt. Dies ist auch innerhalb eines Programms möglich. Es wird durch den Befehl:

PRINT „ ♥ "; erreicht.

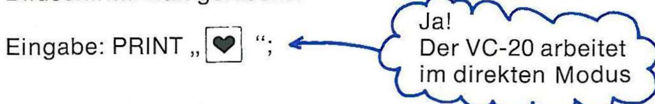
Dieses Zeichen erscheint, wenn innerhalb von Anführungszeichen SHIFT und  gedrückt wird.

Wenn während der Eingabe der PRINT-Nachricht die Tastenkombination SHIFT und CLR HOME betätigt wird, dann erscheint ein Herz in *negativer* Bildschirmdarstellung. Dieses Symbol sagt dem VC-20, daß er den Bildschirminhalt zu löschen und den Cursor auf die *Ausgangsposition* zu plazieren hat. Die Ausgangsposition befindet sich in der oberen linken Ecke des Bildschirms.





Versuchen Sie doch mal, den PRINT-Befehl direkt zu benutzen. Wird der Bildschirminhalt gelöscht?

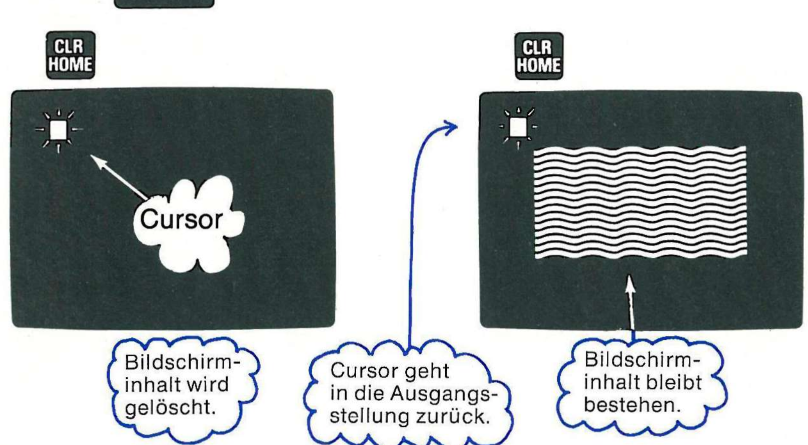


Spielen Sie jetzt ein bißchen auf dem Bildschirm. „Malen" Sie „Vögel", „Sonderzeichen, Buchstaben, Nummern und graphische Symbole ganz willkürlich. Geben Sie nun ein:



Haben Sie den Unterschied zwischen dieser PRINT-Anweisung und der davor benutzten PRINT-Anweisung gemerkt? Dieses mal ging der Cursor zwar in die *Ausgangsposition* zurück, hat jedoch den Bildschirminhalt nicht gelöscht. Wird die SHIFT -Taste während der Betätigung der CLR HOME -Taste *nicht gehalten,* dann erscheint statt des Herzchens ein „S" in negativer Darstellung. Dieses „S" weist den VC-20 an, den Cursor zwar in die *Ausgangsposition* zu bringen, jedoch den Bildschirm nicht zu löschen.

Eingabe: SHIFT



Springender Ball

Löschen Sie jetzt den Bildschirm und geben Sie folgendes ein (RETURN nicht vergessen):

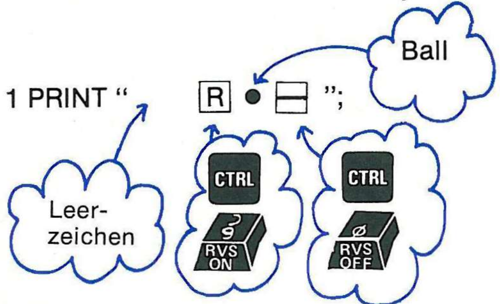


2 GOTO 1

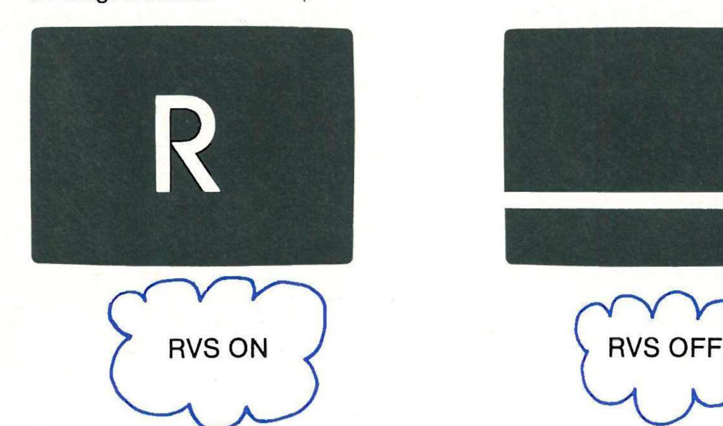
RUN

In diesem Moment wird sich der Bildschirm mit Spalten blauer Bälle auffüllen. Lassen Sie die Bälle für eine Weile vorbeitanzen und drücken Sie dann die RUN STOP -Taste.

Das Drücken dieser Taste hält das Programm an. Dies meldet der VC-20 mit BREAK und READY. Geben Sie jetzt ein:

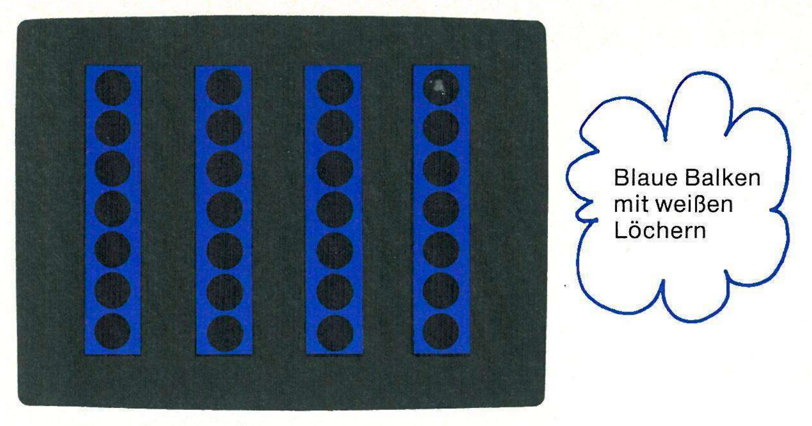


Diese zwei Zeichen, welche die negative oder positive Darstellung steuern, werden vom VC-20 in die Druck-Anweisung mit eingebaut, wenn die CTRL -Taste gehalten wird, und die Tasten RVS ON  und RVS OFF betätigt werden.

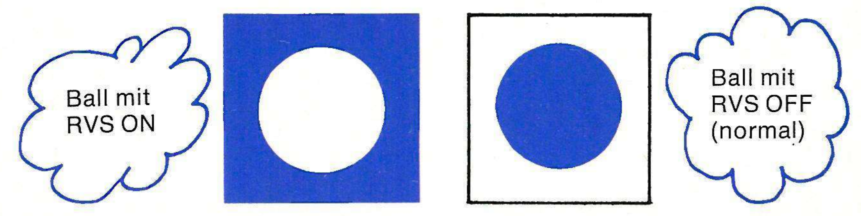


Wurde die neue Zeile eingegeben, veranlassen Sie den VC-20, dieses Beispiel auszuführen. Dafür ist der Befehl RUN einzugeben.

Was passiert eigentlich mit den Bällen? Um das besser ersichtlich zu machen, drücken Sie die CTRL -Taste, womit der Ausgabeprozess verlangsamt wird. Jetzt sollten die Spalten blauer Rechtecke mit weißen Löchern leicht ersichtlich sein. Können Sie sich vorstellen warum?

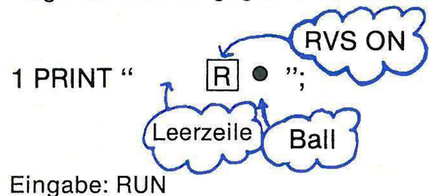


Wenn das Steuerzeichen **RVS ON** vor das Symbol für den Ball plaziert wird, bedeutet das für den VC-20, daß die Farbe *umzukehren* ist, die für die Ballanzeige benutzt wird (wie beim Negativ eines Photos). Bisher war der Ball in blauer Farbe und der unmittelbare Hintergrund weiß. Die Umkehrung ergab einen weißen Ball und einen blauen Hintergrund.



Die Plazierung des Steuerzeichens **RVS OFF** hinter den Ball veranlaßt, daß die Umkehrung des Hintergrunds und für die Zeichenfarbe wieder ausgeschaltet wird. Es ist eigentlich die Umkehrung einer Umkehrung, womit letztlich wieder der Ausgangspunkt erreicht wird. Wird **RVS OFF** nicht eingesetzt, dann resultiert daraus ein sehr interessanter Effekt.

Wenn notwendig, drücken Sie jetzt die RUN STOP -Taste, wonach die folgende Zeile eingegeben werden kann:



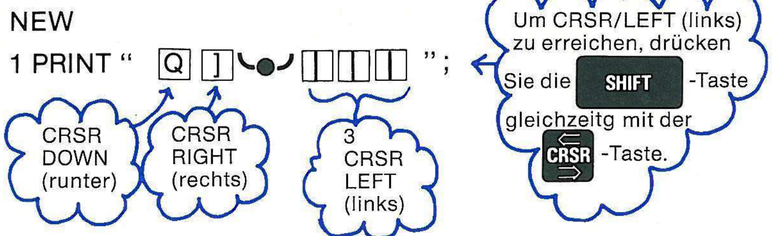
Eingabe: RUN

Wird **RVS OFF** nicht eingesetzt, dann werden alle Leerzeichen auf dem Bildschirm umgekehrt, was zu ganzen Blöcken führt. Daraus ergeben sich sehr interessante Muster, die sich alle sehr leicht mit dem VC-20 erstellen lassen. Wie wäre es, eigene Entwürfe für Stoffe, Tapeten oder Kacheln zu konzipieren? Wir hoffen, damit Ihnen einige Anregungen gegeben zu haben.

Die Cursor-Steuerung

Die Steuertasten für den Cursor ermöglichen es, diesen an jede beliebige Stelle des Bildschirms zu setzen. Es ist auch möglich, die Cursor-Steuerung in einer PRINT-Anweisung einzubetten, was dazu dient, die Zeichen und Nachrichten genau an die Stellen zu bringen, an denen sie gewünscht sind. Das „Vogel"-Beispiel, welches am Anfang dieses Kapitels beschrieben wurde, verfolgt uns bis hierher. Jedoch haben wir noch einige Cursor-Steuerzeichen hinzugefügt.

Löschen Sie den Bildschirm um die Neuauflage der „Vogel"-Serie einzugeben:



In dieser Zeile befinden sich einige Cursor-Steuer-Zeichen: **DOWN** (runter), **RIGHT** (rechts), **LEFT** (links). Damit fliegt der Vogel nicht nur an einer Stelle, sondern bewegt sich auch über den Bildschirm. Wenn Sie alle folgenden Befehle eingeben und danach RUN, dann können Sie diese Bewegung ebenfalls beobachten.

2 FOR T=1 TO 150: NEXT T ← Warten

3 PRINT " ";

3 CRSR Left (links)

Diese Zeile beinhaltet 3 nach **links** ausgerichtete Cursor-Steuerzeichen.

4 FOR T=1 TO 150: NEXT T ← Warten

5 PRINT " "; ← Löschung des Vogels

3 Leerstellen

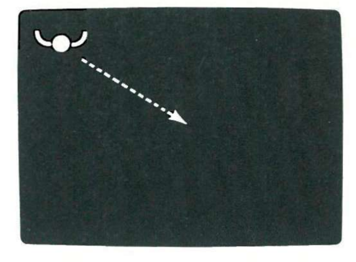
3 CRSR Left links

6 FOR T=1 TO 150: NEXT T ← Nochmal warten

7 GOTO 1

Vergleichen Sie zuerst die Eingabe mit der Vorlage. Ist alles in Ordnung, dann kann mit RUN gestartet werden und die Bewegung des Vogels wird sichtbar.





Die Cursor-Steuerzeichen dieses Beispiels veranlassen den „Vogel" sich nach unten rechts zu bewegen. Übrigens – es ist nicht ganz so leicht zur gleichen Zeit die Melodie „Kommt ein Vogel geflogen" auf dem VC-20 zu komponieren.

Jetzt sind Sie wieder an der Reihe. Wie wäre es mit einem Eigenentwurf?

Zeichentrick-Darstellung mit POKES und PEEKS

Bis zu diesem Punkt wurde beschrieben, wie die PRINT-Anweisung im Zusammenhang mit den Cursor-Steuerzeichen und der Farbselektion eingesetzt werden kann, um damit Objekte in Bewegung zu setzen. Möglicherweise ist dies die schnellste Methode mittels BASIC, Bewegung ins Bild zu bringen. Die meisten Spiele, die denen auf den Spielautomaten gleichen, sind darauf programmiert, daß das bewegende Objekt auf andere sich im Bild befindende Objekte reagiert. Die einzig praktikable Art dies zu programmieren ist die, daß die Befehle PEEK und POKE verwendet werden.

Um diese Technik einzusetzen ist es jedoch notwendig, sich einige Kenntnisse über den speicherbezogenen Bildschirm anzueignen. Jede Bildschirmposition entspricht einer Stelle im Hauptspeicher des VC-20 (RAM). Da der Bildschirm in 506 Positionen aufgeteilt ist, in denen die ausgewählten Zeichen angezeigt werden, sind auch im Hauptspeicher 506 reservierte Stellen vorhanden, welche diesen Bildschirmpositionen zugeordnet sind. Jede Hauptspeicherstelle ist in der Lage, einen numerischen Wert von 0 bis 255 aufzunehmen (siehe Anhang H). Eine sich in einer Hauptspeicherstelle befindende Zahl ist auch gleichzeitig der Schlüssel für ein bestimmtes Zeichen auf der entsprechenden Position.

Unter normalen Umständen ist die erste Bildschirmposition (oben links) mit der Hauptspeicherstelle 7680 und die letzte Bildschirmposition (unten rechts), mit der Hauptspeicherstelle 8185 gleichgesetzt. Genau so wie der Text in einem Buch sind die Stellen von links oben nach rechts unten nebeneinander aufgereiht.

Für dieses Beispiel sollte einmal angenommen werden, daß hier ein hüpfender Ball auf dem Bildschirm zu steuern sei. Der Ball befindet sich augenblicklich in der Mitte des Bildschirms, also Spalte 10 und Zeile 10. (Die obere linke Ecke wird als Spalte 0, Zeile 0 betrachtet.) Die Formel zur Berechnung der entsprechenden Hauptspeicherstelle ist wie folgt:

P = 7680 + X + 22*Y

wobei „X" die Zeile und „Y" die Spalte ist. In diesem Fall bedeutet das, daß sich die Hauptspeicherstelle für die Position des Balles folgendermaßen errechnet: 7680 + 10 + 220 = 7910. Löschen Sie nun den Bildschirm und geben Sie folgenden Befehl ein:

POKE 36879,8

Sie sehen jetzt nur noch schwarz. Aber macht nichts, gleich hellt sich für Sie einiges wieder auf. Geben Sie nun ein:

POKE 7980,81

Der Ball erscheint in der Mitte des Bildschirms. Das entsprechende Zeichen wurde unmittelbar in die entsprechende Hauptspeicherstelle des Bildschirms gebracht (gePOKEd).

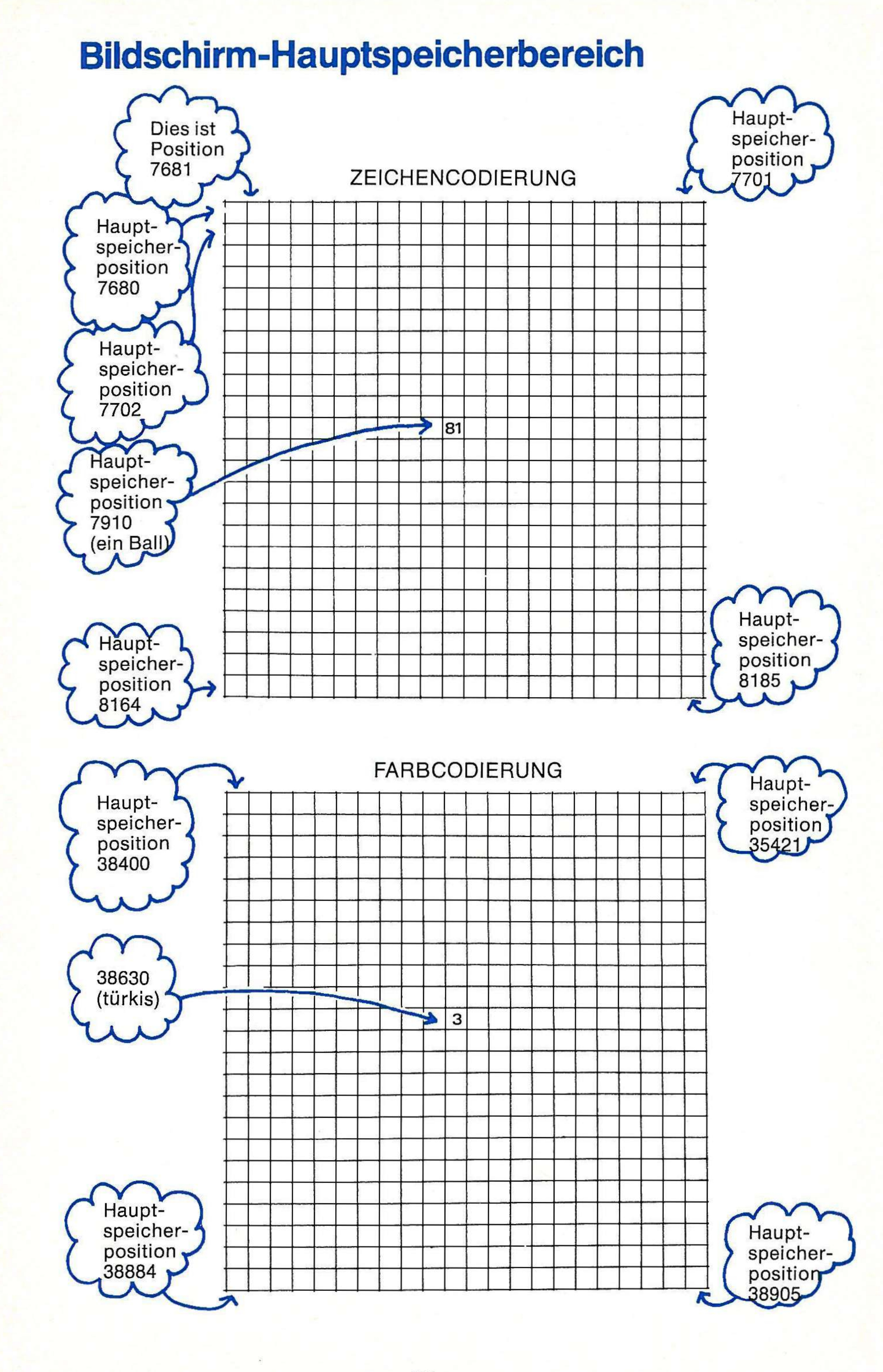
Die Farbe des Balles ist weiß. Es besteht jedoch eine Möglichkeit, die Farbe eines jeden Zeichens zu ändern.

Verwenden Sie dabei ebenfalls einen POKE-Befehl. POKEn Sie also:

POKE 38630,3

Der Ball wird plötzlich türkis. Jeder Bildschirmposition sind im Hauptspeicher 2 Stellen zugeordnet, eine für die Speicherung des Zeichens, und die andere für die Speicherung der Farbe. Der Farbauswahl-Speicher beginnt bei der Hauptspeicherstelle 38400 (obere linke Ecke) und spannt sich über insgesamt 506 Positionen. Dabei erstreckt sich die Farbkodierung von Ziffer 0 bis Ziffer 7, die den 8 Farbtasten (1 bis 8) entsprechen. (Andere Zahlen wirken sich in sehr fremdartigen Resultaten aus. Dazu ist das Programmiererhandbuch des VC-20 einzusehen.)

Bildschirm-Hauptspeicherbereich



Hier ein kurzes Programm, das den Ball auf dem Bildschirm umherspringen läßt. Unten wird dieses Programm detailliert erklärt.

```
10 PRINT " [SHIFT] [CLR HOME] "
20 POKE 36879,9
30 POKE 36878,15
40 X=1
50 Y=1
60 DX=1
70 DY=1
80 POKE 7680+X+22*Y, 81
90 FOR T=1 TO 10: NEXT
100 POKE 7680+X+22*Y, 32
110 X=X+DX
120 IF X=0 OR X=21 THEN DX=-DX: POKE 36876, 220
130 Y=Y+DY
140 IF Y=0 OR Y=22 THEN DY=-DY: POKE 36786, 230
150 POKE 36876, 0
160 GOTO 80
```

Die Zeile 10 löscht den Bildschirm, und die Zeile 20 vergibt die Farben (für den Bildschirm-Hintergrund schwarz und für den Bildschirmrahmen weiß). Die Zeile 30 dreht die Lautstärke voll auf.

Die Variablen „X“ und „Y“ der Zeilen 40 und 50 steuern die Zeile und Spalte, auf welcher sich der Ball befindet. Bei den Werten „DX“ und „DY“ der Zeilen 60 und 70 handelt es sich um die horizontale oder vertikale Richtung der Ballbewegungen. Wenn +1 zu dem Wert in „X“ addiert wird, dann bewegt sich der Ball um eine Spalte nach rechts. Der Ball geht in die andere Richtung, wenn ein Wert von -1 zu „X“ hinzuaddiert wird. Die Addition von +1 auf „Y“ bringt den Ball um eine Zeile nach unten in Bewegung, und die Addition von -1 auf „Y“ bewegt den Ball um eine Zeile nach oben.

Die Zeile 80 bringt das Zeichen für den Ball auf die derzeitige Position des Bildschirms. In der Zeile 90 ist eine Zeitverzögerungs-Schleife, die den Ball gerade so lange auf der entsprechenden Bildschirmposition läßt, daß es für Sie möglich ist, ihn zu sehen. Die Zeile 100 löscht den Ball, indem ein Leerzeichen (Kode 32) an die jeweilige Bildschirmposition gebracht wird.

Die Zeile 110 addiert den Richtungsfaktor zu „X“. Die Zeile 120 überprüft, ob der Ball den seitlichen Bildschirmrahmen erreicht hat und veranlaßt in diesem Fall eine Richtungsänderung und die Abgabe eines kurzen, hörbaren Signals. Die Zeilen 130 und 140 befassen sich mit dem gleichen Thema, jedoch ausschließlich für oberen und unteren Rand.

Die Zeile 150 schaltet den Ton aus, der beim Zusammentreffen von Ball mit Bildschirmrahmen eingeschaltet wurde. Die Zeile 160 verursacht die Rückkehr zum Anfang des Programms, um den Ball wiederum anzuzeigen und an eine weitere Bildschirmposition zu bewegen.

Sie sollten jetzt versuchen, einige Änderungen im Programm vorzunehmen. Indem der Wert „81" der Zeile 80 auf einen anderen Wert gesetzt wird, kann der Ball mit einem anderen Zeichen ausgetauscht werden. Wird entweder DX oder DY auf „0" gesetzt, dann bewegt sich der Ball auf geraden Linien, anstatt diagonal zu verlaufen.

Jetzt ist es an der Zeit, etwas mehr Intelligenz in das Programm einzubauen. Bis jetzt wurde nur überprüft, ob die Werte „X" und „Y" an den Bildschirmrahmen gelangen. Wir schlagen vor, daß die folgenden Zeilen dem vorhergehenden Programm hinzugefügt werden. Dafür ist lediglich notwendig, diese Zeilen so wie sie hier dargestellt sind einzugeben. Sie werden dem Programm automatisch hinzugefügt.

```
 32 FOR L=1 TO 10
 35 POKE 7680 + INT (RND (1) *506), 102
 37 NEXT
155 IF PEEK (7680 + X + 22*Y) = 102 THEN DX = -DX: DY = -DY:
    POKE 36876, 180: GOTO 110
```

Die Zeilen 32 bis 37 setzen 10 graue Zeichen in willkürliche Positionen des Bildschirms. Die Zeile 155 überprüft („PEEKs"), ob der Ball gerade im Begriff ist, in einen dieser grauen Blöcke zu springen. Ist das der Fall, dann wird die Bewegungsrichtung ebenfalls geändert.

Für weitere Informationen im Zusammenhang mit der Zeichentrick-Darstellung lesen Sie die Anhänge H und I.



Die musikalische Stunde

- Musizieren mit dem VC-20
- Die 4 Stimmen des VC-20
- Tabelle der Noten
- Geräuschkomposition
- Der VC-20 als Klavier
- Spielen von Melodien
- Einiges über „POKE“

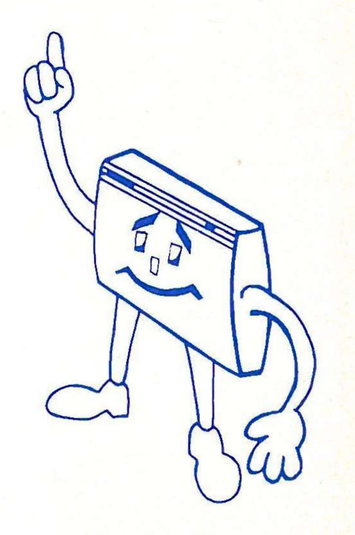
Programmvorschlag

Schreiben Sie dieses Programm genau ab und sehen Sie mal, was passiert!

```
NEW
 10 PRINT " SHIFT CLR HOME "
 20 FOR I = 1 TO 5
 30 POKE 36873 + I, 0
 40 NEXT I
 50 PRINT "WELCHE STIMME (1-4)?"
 60 PRINT "ENDE?, DANN Ø"
 70 INPUT N
 80 IF N = 0 THEN END
 90 PRINT "WELCHER TON (128-254)?"
100 INPUT P
110 POKE 36878, 4
120 POKE 36873 + N, P
130 FOR J = 1 TO 2000: NEXTJ
140 GOTO 10
```

Um das Programm anzuhalten, drücken Sie die 

-Taste.



Musizieren mit dem VC-20

Sie sind sich bis jetzt sicherlich noch nicht der Tatsache bewußt gewesen, daß es mit dem VC-20 möglich ist zu musizieren und Spezialtoneffekte zu erzeugen. Dieses Kapitel gibt eine Einführung in die Möglichkeiten des Musizierens und des Aufbaus von Spezialgeräuschen. Es ist ein Kurzlehrgang in der Steuerung von Tönen und dem Abspielen jeder Art von Musik, angefangen von Bach bis hin zum Rock. Je nach Gemütsverfassung bleibt es Ihnen jetzt überlassen, sich in romantisches Kerzenlicht oder schillerndes und blinkendes Disco-Spot-Licht einzuhüllen.

Zuerst ist es notwendig zu erkennen, wie eine Note überhaupt gespielt werden kann:

Dazu ist zuerst die SHIFT -Taste zu halten und die CLR HOME -Taste zu betätigen.

N E W und Drücken der RETURN -Taste.

P O K E SPACE 3 6 8 7 8 , 3

und Betätigung der RETURN -Taste.

P O K E SPACE 3 6 8 7 5 , 2 2 5

und Betätigung der RETURN -Taste.

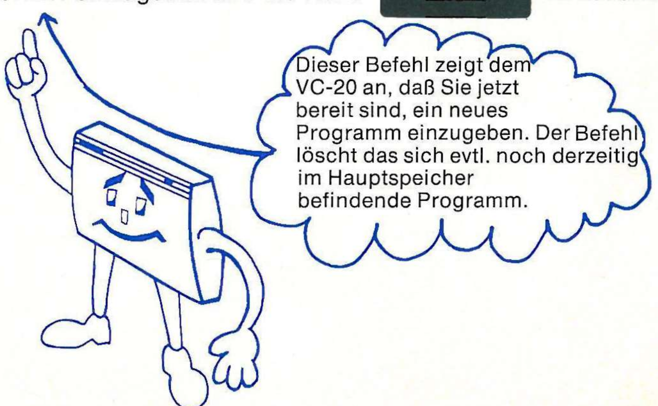
Jetzt sollte ein Ton aus dem Lautsprecher des Fernsehgeräts erklingen. Ist dies nicht der Fall, dann sollte der Lautstärken-Regler des Fernsehgeräts auf die übliche Lautstärke gesetzt werden.

Es wurde gerade das Mittel-C gespielt, eine der 128 Noten des gesamten Repertoires des VC-20. Um diesen Ton wieder auszuschalten, geben Sie ein:

P O K E SPACE 3 6 8 7 5 , 0

Finden Sie doch mal heraus, wie viele Oktaven der VC-20 spielen kann:

Dazu ist NEW einzugeben und die Taste RETURN zu drücken:



Schreiben Sie nun dieses Programm:

1 POKE 36878, 15 RETURN

2 FOR I = 128 TO 255 RETURN

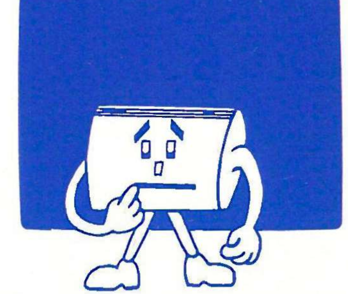
3 POKE 36875, I RETURN

4 FOR D = 1 TO 100 RETURN

5 NEXT D RETURN

6 NEXT I RETURN

7 POKE 36875, 0 RETURN



Vergleichen Sie bitte genau die eingegebenen Informationen mit der Vorlage. Ist bei der Eingabe ein Fehler unterlaufen, kann die entsprechende Zeile schlicht und einfach überschrieben werden. Dazu ist es selbstverständlich notwendig, die Zeilennummer der fehlerhaften Zeile einzugeben. Der VC-20 ersetzt automatisch die alte Zeile mit der Neueingabe. Um eine Auflistung des Programms zu erhalten, ist:

LIST einzugeben und RETURN zu drücken.

Sind alle Zeilen in Ordnung, dann

RUN eingeben und RETURN drücken.

Jetzt ertönen alle 128 Noten, die der VC-20 erzeugen kann. (Ist das nicht der Fall, dann sind die eingegebenen Befehle zu überprüfen.) Eine Wiederholung dieser Tonleiter wird durch die Eingabe von:

RUN und RETURN erreicht.

Da Sie sich jetzt offensichtlich mit der Stimmlage des VC-20 angefreundet haben, ist es langsam an der Zeit, daß einige Erklärungen darüber abgegeben werden.

Im Inneren des VC-20 steuert der Video-Interface-Chip sowohl Ton als auch Bild. Dieser Baustein ermöglicht es dem VC-20, Töne an den Lautsprecher des angeschlossenen Fernsehgeräts weiterzuleiten. Die Lautstärke kann sowohl vom Lautstärkenregler des Fernsehgeräts als auch über die VC-20-Tastatur gesteuert werden.

Die vier Stimmen des VC-20

Der VC-20 verfügt über vier Stimmlagen. Er ist also in der Lage, vier Stimmen zur gleichen Zeit zu produzieren. Sie können sich diese als „Sopran", „Alt", „Tenor" und „Geräusch" vorstellen. Jede dieser Stimmen hat eine bestimmte „Stimmen-Steuerungsnummer". Unter Verwendung dieser Nummer kann der Lautsprecher „eingeschaltet" werden, wonach eine Stimme oder ein Geräusch aus dem Lautsprecher ertönt. Dafür benutzen Sie den POKE-Befehl.

Die vier „Stimmen-Steuerungsnummern" (Adressen) des VC-20 sind:

36874 (Stimme 1 – Musik)
36875 (Stimme 2 – Musik)
36876 (Stimme 3 – Musik) und
36877 (Stimme 4 – Geräusch).

Die Steuerungsnummer (Adresse) für die Lautstärke ist 36878.



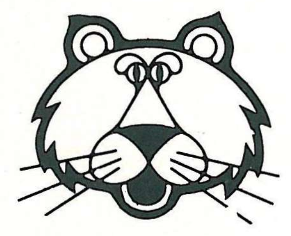
36874 . . . (Alt)



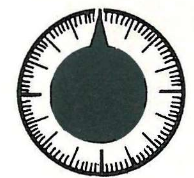
36875 . . . (Tenor)



36876 . . . (Sopran)



36877 . . . (Geräusch)
Stimme 4



36878 . . . (Lautstärke)

Um eine Note zu spielen, ist der Befehl POKE, die Stimmen-Steuerungsnummer, ein Komma und die Nummer der entsprechenden Note einzugeben.

Zum Beispiel:

POKE 36874, 128 und RETURN drücken.

Nun ertönt die Note # 128 unter Verwendung der Stimme # 36874 (der niedrigsten Stimme). Bis zum Ende dieses Kapitels werden jetzt Kurzbezeichnungen für die einzelnen Stimmlagen wie folgt benutzt:

niedrigste Stimme = S1
mittlere Stimme = S2
höchste Stimme = S3
Sonderstimme = S4 (mit Geräusch?!)
(Wird für Spezialtoneffekte eingesetzt).

Dem VC-20 geht nie die Luft aus. Daher ist es notwendig, daß er mit einem entsprechenden Befehl zum Schweigen gebracht wird, und zwar:

POKE 36874,0 und RETURN drücken.

Wenn Sie noch kein voll ausgebildeter Tontechniker sind, könnte diese Prozedur etwas kompliziert sein. Aus diesem Grund haben wir eine etwas angenehmere Lösung geschaffen. Es ist möglich, sich viele der normal notwendigen Eingaben zu ersparen, wenn das folgende Kurzprogramm eingesetzt wird, welches die Kurzbezeichnungen der Stimmlagen-Nummern übersetzt.

NEW

S1 = 36874 RETURN

S2 = 36875 RETURN

S3 = 36876 RETURN

S4 = 36877 RETURN

L = 36878 RETURN

Dies ermöglicht es Ihnen, die Stimmlagen-Nummern durch die Kurzbezeichnungen S1, S2, S3 und S4 und die Lautstärke durch die Kurzbezeichnung L einzugeben. Die eingegebenen Informationen sollten jetzt überprüft werden. Hat alles seine Richtigkeit, dann sind Sie jetzt in der Lage, ohne große Anstrengungen Musikstücke zu erstellen.

POKE L, 10 RETURN

Dieser Befehl stellt die Lautstärke ein.

Der Wert 10 wird in die Hauptspeicherstelle (L) gebracht, welche die Lautstärke steuert. In dieser Position können Werte von 0 bis 15 gespeichert werden. Je höher die Nummer, umso lauter sind die Töne.

POKE S1, 195
POKE S2, 215
POKE S3, 231

Experimentieren Sie jetzt ein wenig mit dem Einsatz dieser Methode. Sind Sie der Töne überdrüssig (und schließlich haben andere Leute ja auch ab und zu etwas zu sagen), dann kann der VC-20 zum Schweigen gebracht werden, indem der Wert 0 den Stimmlagen-Nummern zugeordnet wird. Dies sieht dann wie folgt aus:

POKE S1, 0
POKE S2, 0
POKE S3, 0

Es folgt eine Tabelle der Werte, die den einzelnen Stimmlagen zur Erreichung der unterschiedlichen Noten zugeordnet werden können. (**Anmerkung:** Zahlen kleiner als 128 erzeugen Stille):

Tabelle der Noten

NOTE	WERT	NOTE	WERT
C	135	G	215
C#	143	A^b	217
D	147	A	219
E^b	151	B	221
E	159	H	223
F	163	C	225
F#	167	C#	227
G	175	D	228
A^b	179	E^b	229
A	183	E	231
B	187	F	232
H	191	F#	233
C	195	G	235
C#	199	A^b	236
D	201	A	237
E^b	203	A#	238
E	207	B	239
F	209	C	240
F#	212	C#	241

Durch die Eingabe (POKE) von Werten in die ersten drei Stimmlagen (S1 – S3) ist es sogar möglich, ganze Musikstücke abzuspielen. (Die Erzeugung von Geräuscheffekten [S4] ist in diesem Zusammenhang nicht besonders wünschenswert, es sei denn, daß Sie beabsichtigen,

demnächst mit Pink Floyd aufzutreten.) Leider ist diese POKE-Vorgehensweise für den Menschen sehr mühsam . . . , jedoch für den VC-20 ein Kinderspiel. Daher sollte man noch einen Schritt weiter gehen und diese POKEs in ein Programm einbetten, das dem Computer die Arbeit überträgt. Wie wäre es mit „Für Elise" oder „Santa Lucia"?

Geräuschkompositionen

Die vierte Stimme, S4, wird mit 36877 adressiert. Hierbei handelt es sich um einen Spezialeffekt-Generator. So z. B. ist es möglich, die Geräusche eines Flugzeugs nachzuahmen:

```
POKE L, 6
POKE S4, 130
```

ein sehr niedriger Ton

Es ist eine 4-Propeller-Maschine – nicht war? Wie sieht es jetzt mit ein wenig Wind aus, der um die Segel eines Segelflugzeuges streicht?

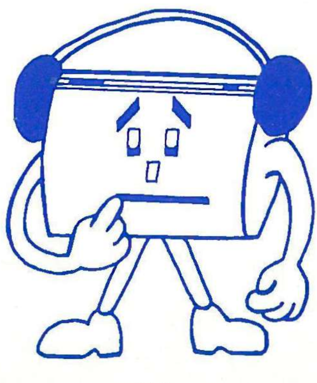
```
POKE S4, 240
```

Nicht vergessen, die Lautstärke einzuschalten:

```
POKE L, 4
```

oder was für eine Lautstärke Sie auch immer bevorzugen. So weit, so gut. Wollen wir den Ton abschalten?

```
POKE S4, 0
```



Um bessere Möglichkeiten zu haben, geben Sie die folgenden DATA-Anweisungen ein. Es handelt sich hierbei um eine altbekannte Melodie:

```
300 DATA 225, 360, 225, 360, 225, 240
310 DATA 228, 120, 231, 360, 231, 240
320 DATA 228, 120, 231, 240, 232, 120
330 DATA 235, 720, 240, 360, 235, 360
340 DATA 231, 360, 225, 360, 235, 240
350 DATA 232, 120, 231, 240, 228, 120
360 DATA 225, 480
370 DATA -1
```





Oder wenn Sie die Klassiker bevorzugen:

```
300 DATA 217, 400, 213, 400, 223, 400
310 DATA 227, 200, 234, 200, 230, 400
320 DATA 227, 200, 234, 200, 230, 400
330 DATA 223, 400, 227, 400, 217, 400
340 DATA 213, 600, -1
```

Für das Abspielen längerer Stücke können natürlich mehr Data-Anweisungen eingegeben werden. Für unseren Zweck sollten Sie sich auf drei Noten (= 6 Zahlen) je Zeile beschränken.

Um diese Melodien abspielen zu können, müssen Sie noch die nötigen Anweisungen programmieren. Wollen Sie es mal allein probieren? – Viel Glück, Herr Mozart jun. – (Auf Seite 77 finden Sie in den Zeilen 10 bis 220 die optimale Lösung.)

Der VC-20 als Klavier

Als letztes stellen wir hier noch das Programm vor, welches es Ihnen ermöglicht, die Tastatur des VC-20 als Klaviatur einzusetzen:

```
NEW

 10 REM STIMMLAGENREGISTER
 15 REM SPEICHERN
 20 S2 = 36875
 30 L  = 36878
 40 POKE S2, 0

100 REM H-DUR SPEICHERN
110 FOR N = 1 TO 8
120 READ A (N)
130 NEXT N

140 DATA 223, 227, 230
150 DATA 231, 234, 236
160 DATA 238, 239
```

Schafft Abkürzungen für die benötigten Stimmlagen-Register und schaltet diese aus.

Liest H-Dur von Zeilen 140-160.

Enthält POKE-Werte für H-Dur

```
200 REM AUF TASTATUR SPIELEN
210 POKE L, 3

220 GET A$: IF A$ = " " THEN 220
230 N = VAL (A$)

240 IF N = O OR N = 9 THEN 300

250 POKE S2, 0
260 FOR T = 1 TO 25: NEXT T

270 POKE S2, A (N)
280 GOTO 220

300 REM MODULENDE
310 POKE S2, 0
```

Schaltet Lautstärke ein

Erkennt, welche Taste gedrückt wurde

Beendet das Programm, wenn „0“ oder „9“ gedrückt wurde

Kurze Pause zwischen den Noten

Spielt den Ton und geht zurück in Erwartung weiterer Eingaben

Ausschalten der Lautstärke

Wenn Sie jetzt den Befehl RUN eingeben (mit RETURN), ist es möglich, Melodien auf dem VC-20 zu spielen. Die Tasten der obersten Reihe, die mit Zahlen beschriftet sind, können jetzt zum „musizieren“ eingesetzt werden:

****

1	2	3	4	5	6	7	8
A	H	CIS	D	E	FIS	GIS	A

****

Der VC-20 wird so lange eine Note spielen, bis eine andere Note angeschlagen wurde. Wenn Sie genug gespielt haben, geben Sie entweder „0“ oder „9“ ein und das Programm wird beendet. Ihren Zuhörern können Sie eine „Zugabe“ gewähren, wenn jetzt nochmals der Befehl RUN eingegeben wird.

Sie sollten einmal die folgenden Tasten anschlagen:

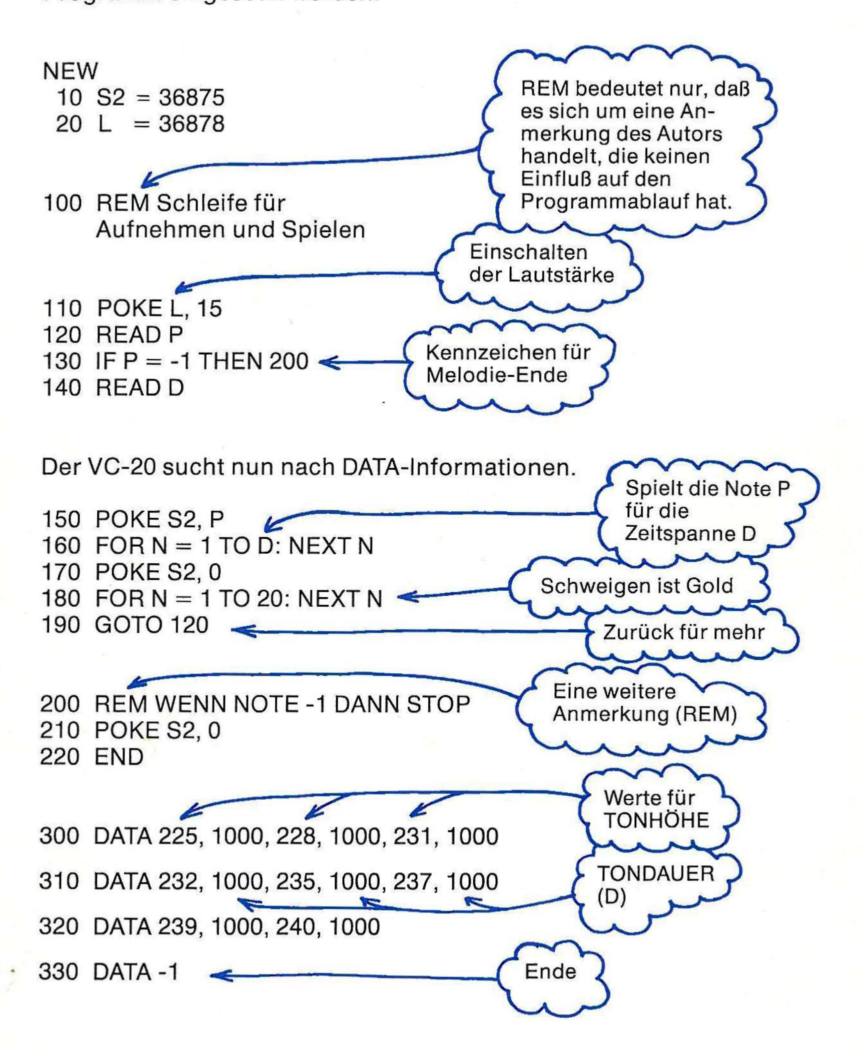
```
1 1 5 5 6 6 5
4 4 3 3 2 2 1
5 5 4 4 3 3 2
5 5 4 4 3 3 2
1 1 5 5 6 6 5
4 4 3 3 2 2 1 8
9
```

ODER:

```
3 3 4 5 5 4 3 2
1 1 2 3 3 2 2
3 3 4 5 5 4 3 2
1 1 2 3 2 1 1
0
```

Spielen von Melodien

Unter Verwendung der Stimmlagen des VC-20 und einer Tabelle der Noten ist jetzt jedem möglich, eigene Melodien zu komponieren oder Melodien aus einem Liederbuch zu übertragen. Dazu sollte das folgende Programm eingesetzt werden:




```
NEW
 10 S2 = 36875
 20 L  = 36878

100 REM Schleife für
    Aufnehmen und Spielen

110 POKE L, 15
120 READ P
130 IF P = -1 THEN 200
140 READ D
```

Der VC-20 sucht nun nach DATA-Informationen.

```
150 POKE S2, P
160 FOR N = 1 TO D: NEXT N
170 POKE S2, 0
180 FOR N = 1 TO 20: NEXT N
190 GOTO 120

200 REM WENN NOTE -1 DANN STOP
210 POKE S2, 0
220 END

300 DATA 225, 1000, 228, 1000, 231, 1000
310 DATA 232, 1000, 235, 1000, 237, 1000
320 DATA 239, 1000, 240, 1000
330 DATA -1
```

In der ersten Befehlsgruppe wurden zuerst die POKE-Stellen angesprochen, die im Zusammenhang mit den Stimmlagen Verwendung finden (in diesem Fall Stimmlage 2 und Lautstärke). Außerdem wurden auch die Kurzbezeichnungen definiert.

Die Zeile 100 enthält eine Anmerkung (REM), die erklärt, was in diesem Teil des Programms geschieht. Es handelt sich hierbei um eine sogenannte „Schleife", da dieser Teil des Programms einen Ton aufnimmt, diesen spielt und dann wieder zum Anfangspunkt des Programms zurückkehrt, in der Erwartung, eine weitere Note vorzufinden. Die Zeile 110 dreht die Lautstärke auf.

Jetzt werden dem VC-20 durch die DATA-Zeilen mitgeteilt, welche Noten zu spielen sind.

VC-20-Tip:

Es ist nicht notwendig, daß die Zeilen mit der Nummer 1 beginnen und in Schritten von 1 eingegeben werden. Die meisten Programme fangen mit 10 an, wobei jeder weitere Programmschritt um 10 erhöht wird. Das ermöglicht es, daß auch Zeilen zwischen die ursprünglich eingegebenen Befehle gesetzt werden können: z. B. 11, 12, usw. zwischen die Zeilen 10 und 20.

Die Zeile 120 weist den VC-20 an, Informationen zu lesen – genannt P – um festzustellen, welche Note zu spielen ist. Diese Information ist in der DATA-Zeile enthalten, mit der wir uns bis jetzt noch nicht befaßt haben. Auf ähnliche Art und Weise liest der Befehl der Zeile 140 eine Information – genannt D –, die eine Aussage über die Länge der Note macht.

Besondere Aufmerksamkeit ist der Zeile 130 zu geben. Die Funktion dieser Zeile veranlaßt die Beendigung des Programms nach dem Lesen der letzten Note. Ohne ein Ende-Markierungszeichen würde das Programm versuchen Noten zu lesen, die gar nicht geschrieben wurden und einen Fehler verursachen. Die Zeile 130 gibt dem VC-20 die Anweisung, daß im Falle der Eingabe eines Wertes von -1 die Ende-Markierung erreicht ist und daß das Programm zur Endroutine zu springen hat. Es ist deswegen notwendig, ein -1 an das Ende der letzten DATA-Zeile zu plazieren.

Jetzt ist der VC-20 tatsächlich in der Lage, Noten zu spielen, den Ton zu unterbrechen und festzustellen, ob eine weitere Note angeschlagen wurde.

Die Zeile 150 bringt lediglich die in Zeile 120 aufgenommene Note in die Stimmlage 2, währenddem die Zeile 160 eine Zeitverzögerung verursacht, bevor die Zeile 130 wieder zum Zuge kommt. Ähnlich ist es mit den Zeilen 170 und 180, in denen die Stimmlage 2 für einen kurzen Moment ausgeschaltet wird. Die Zeile 190 veranlaßt den Rücksprung auf die Zeile 120, wo wieder die nächste Note angenommen wird.

Nun wenden wir uns der Ende-Routine zu. Hierbei sollten Sie sich daran erinnern, daß das Programm bei Erkennen des Ende-Kennzeichens zur Zeile 200 springt. Die Zeile 210 schaltet den Ton aus und die Zeile 220 gibt dem VC-20 zu erkennen, daß das Programm beendet ist.

Obwohl damit die Endroutine abgehandelt ist, ist das Programm jedoch nicht ganz fertig. Das Programm muß noch immer mit den geheimnisvollen Anweisungen ausgestattet werden, die bestimmen, welche Noten zu lesen (READ) sind. Dabei handelt es sich um DATA-Anweisungen, die so genannt werden, weil sich hinter ihnen bestimmte Daten verbergen. Es ist dabei unerheblich, an welchen Stellen innerhalb des Programms sich diese DATA-Anweisungen befinden. Wann auch immer der VC-20 einen READ-Befehl erhält, sieht er sich nach einer DATA-Anweisung um.

Dieses Programm enthält die Daten für C-Dur. Die erste Zahl ist der POKE-Wert für die erste Note, nämlich 225 – C. Die zweite Zahl setzt die Notenlänge, nämlich 1000 (der VC-20 zählt von 1 bis 1000) – ungefähr eine Sekunde. Die Zeile 310 beherbergt die nächsten drei Noten und die Zeile 320 die letzten zwei Noten. Die hier stehenden Werte wurden der vorhergehenden Tabelle entnommen.

Nach einer Überprüfung der eingegebenen Befehle kann RUN eingegeben werden. Danach sollte eine ziemlich genaue Tonleiter in C-Dur erklingen. Sind trotzdem ab und zu „schräge" Töne zu hören, sollten Sie versuchen, die Werte der DATA-Anweisungen etwas zu ändern.

Also nochmal: Es ist möglich, DATA-Anweisungen an beliebige Stellen innerhalb des Programms einzusetzen. Das Programm wird jede einzelne DATA-Anweisung lesen, wobei mit der niedrigsten Zeilennummer angefangen wird und nach und nach bis zur letzten DATA-Anweisung alle einzeln abgehandelt werden.

Einiges über „POKE“

Der Befehl POKE ermöglicht es Ihnen, den VC-20 auf einer völlig neuen Ebene zu programmieren. Der Einsatz des Befehls POKE erlaubt es, sich unmittelbar an eine bestimmte Hauptspeicher-Stelle zu wenden und den Inhalt derselben zu ändern. Da dieser Befehl eine direkte Auswirkung auf den Inhalt des Hauptspeichers des VC-20 hat, ist es durchaus möglich, daß Fehler dadurch geschehen, daß die falschen Werte in die richtigen Stellen oder richtige Werte in die falschen Stellen gebracht werden. An dieser Stelle ist es angebracht das zu wiederholen, was schon im ersten Kapitel sehr deutlich ausgesprochen wurde: es *besteht absolut keine Möglichkeit, den VC-20 zu beschädigen...* auch nicht mit einem POKE-Befehl. Aber lassen Sie bitte dem VC-20 seinen Stolz! Bei bestimmten Situationen, die durch einen Fehler des Benutzers verursacht wurden, könnte sich der VC-20 stillschweigend zurückziehen und keinen Ton mehr von sich geben, geschweige denn ein Bild. Wenn Sie mit diesem Teil sowieso fertig sind, dann empfehlen wir Ihnen als Beispiel, den folgenden Befehl einzugeben:

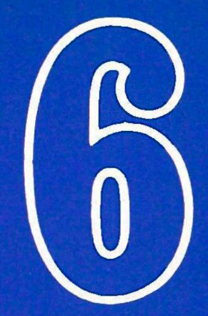
POKE 788, 0

Sie sehen, das auf diese Weise „beleidigte“ System ist nur noch mit der Betätigung der RUN/STOP-Taste und der RESTORE-Taste wieder in Gang zu bringen. Es gibt sogar Fälle, in denen es notwendig ist, die Anlage vollkommen auszuschalten, und dann wieder einzuschalten (einige Sekunden warten). Der Computer wurde dadurch nicht beschädigt. Es bedeutet jedoch, daß das jeweilige Programm, an welchem gerade gearbeitet wurde, verloren gegangen ist. Der Verlust besteht einzig und allein darin, daß die schon einmal eingegebenen Befehle nochmals einzugeben sind (mit der Ausnahme selbstverständlich des fatalen Fehlers).

Trotz allem soll dies ein Hinweis dazu sein, daß der Einsatz der POKE-Befehle mit Sorgfalt zu wählen ist. Es unterstreicht außerdem den Wert von Zusatzgeräten wie Disketten-Laufwerke und Bandkassetten, auf denen Programme gespeichert werden können, sowie Druckern, mit denen zumindest das jeweilige Programm aufgelistet werden kann, was bei der Neueingabe des Programms behilflich ist.

Wir empfehlen Ihnen deswegen, sich die Erklärungen im Zusammenhang mit dem POKE-Befehl im Anhang C näher anzusehen, bevor er im größeren Stil zum Einsatz kommt. Die größeren Nummern sollten sehr vorsichtig eingegeben und vor dem Start durch RUN eingehend überprüft werden. Ein Computer, welcher mit falschen Werten gePOKEd wurde, kann sehr leicht „tot“ spielen.

Dieses Kapitel gab eine Anleitung, die vielleicht Bach und Beethoven aufhorchen ließ (oder auch nicht). Des weiteren ist es jetzt möglich, Freunde und die ganze Familie zum Irrsinn zu treiben und auch zu dem Wunsch, daß die Tage zurückkehren mögen, an denen Computer dieser Art sehr bescheiden und leise waren. Was macht es schon aus, wenn der VC-20 nicht ganz so talentiert wie ein ausgewachsener Konzertflügel ist? Sie und der VC-20 können sich immerhin noch prächtig mit den verschiedenen musikalischen Einsatzmöglichkeiten amüsieren!

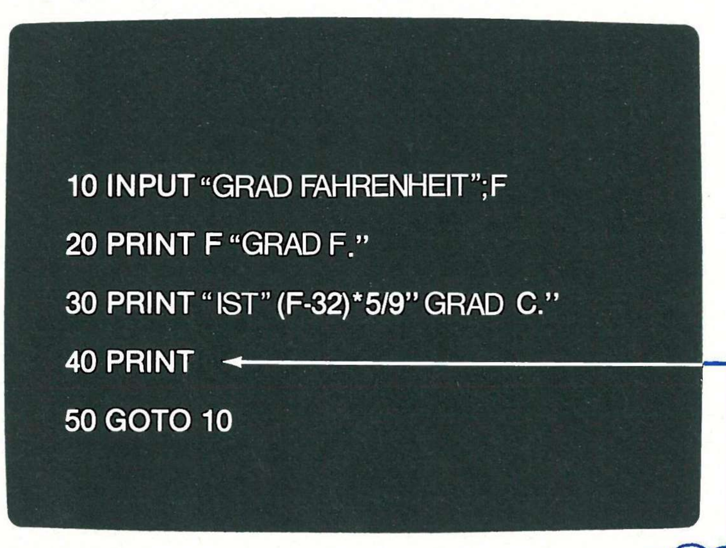


Konversation mit dem VC-20

- Wie heißen Sie?
- Einführung in die Datenfelder
- Auswahl einer Note
- Die GET-Anweisung

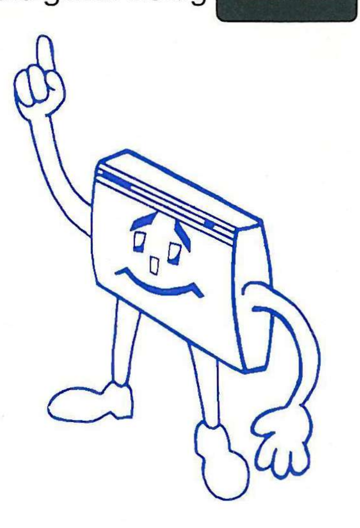
Programmvorschlag

Schreiben Sie nun folgendes Programm genau wie angegeben und schauen Sie was passiert:



Die Eingabe einer *PRINT*-Anweisung ohne Zusatz erzeugt eine Leerzeile auf dem Bildschirm, wenn das Programm ausgeführt wird. Probieren Sie es mal mit und ohne diese Zeile.

Halten Sie das Programm durch **RUN STOP** an und drücken Sie gleichzeitig **RESTORE** .



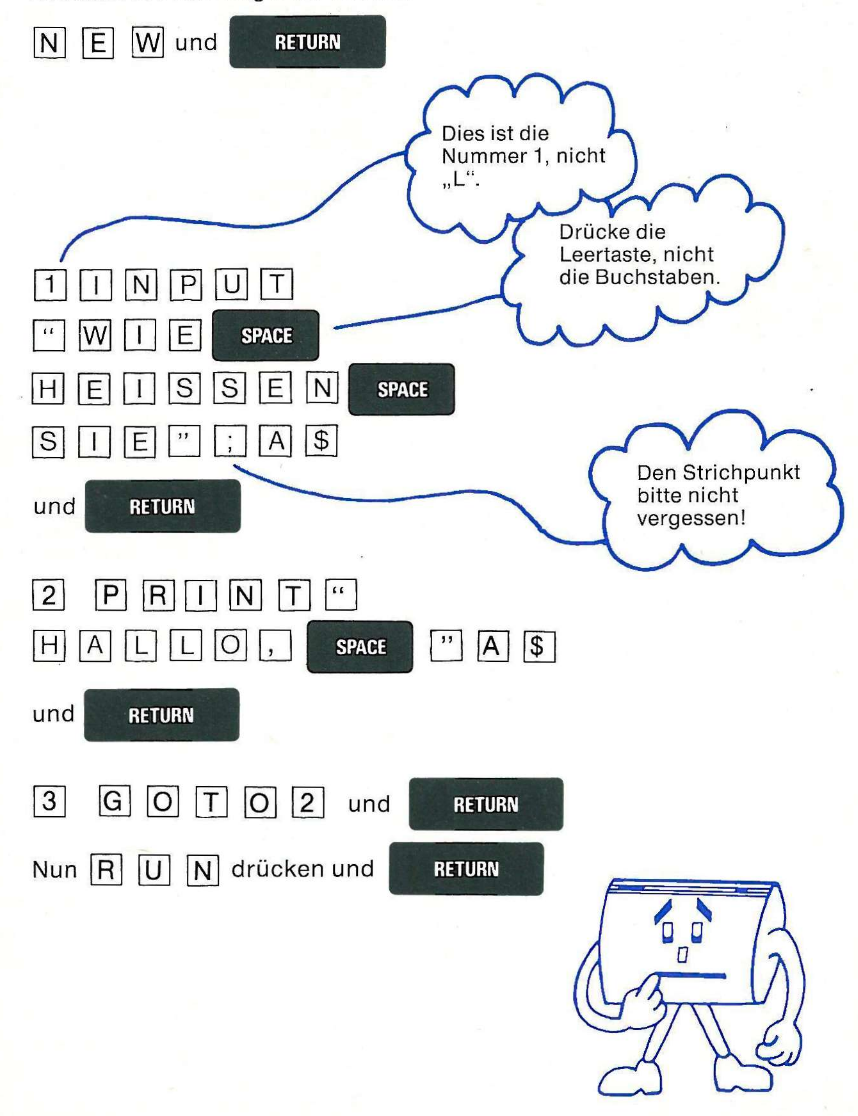
Wie heißen Sie?

Der VC-20 wird nun für uns einige Wunder vollbringen. Wir sind uns sicher, daß er dies auch kann!

Ihr VC-20 ist jetzt empfangs- und betriebsbereit, Ihren Anweisungen zu folgen, wie dies bisher noch nie ein Computer für Sie vollzogen hat.

Zuerst – wie üblich – die CLR HOME -Taste in Kombination mit der SHIFT -Taste drücken.

Drücken Sie nun folgende Tasten:



Der VC-20 antwortet:

```
1 INPUT "WIE HEISSEN SIE";A$
2 PRINT " HALLO, " A$
3 GOTO 2
RUN
WIE HEISSEN SIE ?
```

Beachten Sie, daß der Computer automatisch ein Fragezeichen setzt, wenn er auf Eingaben (INPUT) von Ihnen wartet.

Geben Sie jetzt Ihren eigenen Namen ein und RETURN .

Siehe da, Ihr Name tanzt über den ganzen Bildschirm. Um diesen Vorgang in seiner Geschwindigkeit zu reduzieren, drücken Sie lediglich die CTRL -Taste. Wenn Sie genug davon haben, Ihren Namen im Fernsehen zu sehen, dann drücken Sie einfach RUN STOP . Selbstverständlich nimmt der VC-20 auch andere Namen an. Dazu ist lediglich der Befehl RUN *nochmals* einzugeben, wonach der VC-20 wiederum um die Eingabe eines Namens bittet.

Diese Aufforderung zur Eingabe (INPUT) von Informationen sind für den VC-20 sehr wichtig. Der Computer hat nur begrenzte Möglichkeiten, mit der er Informationen von der Außenwelt erhalten kann. Eine der vielleicht nützlichsten Methoden zur Sammlung von Informationen über die Tastatur wird durch die INPUT-Anweisung (EINGABE) gesteuert. Es ist jetzt wiederum an der Zeit, sich die einzelnen Schritte des eingegebenen Programms etwas genauer anzusehen.

Folgendes wurde dem VC-20 aufgetragen:

Als **erstes** bringe die Nachricht „WIE HEISSEN SIE“ auf den Bildschirm und warte auf die EINGABE (INPUT) von Zeichen über die Tastatur. Speichere diese Eingabe in der Variablen „A$“ (sprich A-String).

Als **zweites** schreibe „HALLO“, gefolgt von dem eingegebenen Namen (aus A$).

Drittens, gehe zurück zur Zeile 2.

Durch die Kombination der zweiten und dritten Zeile wurde der VC-20 veranlaßt, die Nachricht „HALLO“ und den „Namen“ kontinuierlich auf den Bildschirm zu bringen. Ändert man die Zeile 3 in:

3 GOTO 1

dann erscheint die entsprechende Nachricht nur einmal auf dem Bildschirm und der VC-20 wird direkt danach wieder nach einem Namen fragen. Diese Änderung verursacht eine Art von Gedächtnisverlust beim VC-20.

Wann immer auch eine INPUT-Anweisung im Programm gefunden wird, hält dieses an und wartet auf die Eingabe des Benutzers. An dieser Stelle ist es sehr wichtig zu erwähnen, daß der VC-20 eine sehr geduldige Anlage ist und wenn nötig, bis in alle Ewigkeit wartet (oder zumindest so lange, bis die RETURN -Taste betätigt wurde).

In dem Programmbeispiel wurde eine höfliche Art der Fragestellung ausgewählt. Haben Sie nicht eine entsprechende Frage programmiert, dann stellt der VC-20 nur ein Fragezeichen (?) hin. Um keine Verwirrungen zu erzeugen empfehlen wir, die Aufforderung zur Eingabe von Daten immer von einer entsprechenden Frage zu begleiten.

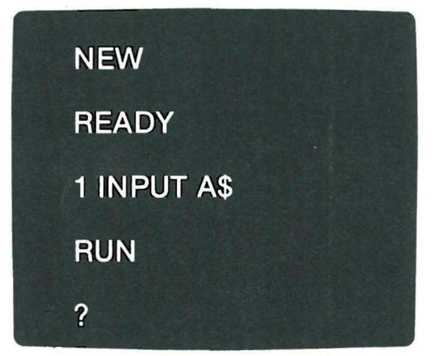
Drücken Sie RUN STOP und RESTORE nacheinander.

Eingabe
von NEW und RETURN

1 INPUT A$

Wenn wir jetzt RUN eingegeben, sieht der Bildschirm wie folgt aus:

Das Fragezeichen allein erzeugt mehr Fragen als beantwortet werden können.



Deswegen sollte jetzt folgendes ausprobiert werden:

1 PRINT „WIE HEISSEN SIE BITTE": INPUT A$

2 PRINT „WAS IST IHR LEIBGERICHT": INPUT B$

3 PRINT

4 PRINT „DANKE „A$" FÜR DIE HÖFLICHE ANTWORT"

5 PRINT „ICH SERVIERE IHNEN ETWAS SPACE „B$"
SPACE ALS BEWEIS MEINER DANKBARKEIT"

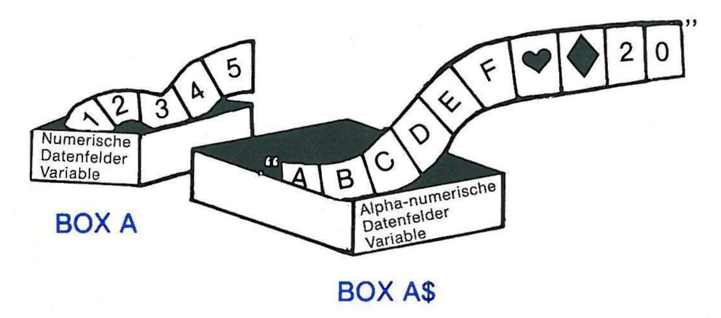
Nachdem der Befehl RUN eingegeben und RETURN -Taste gedrückt wurde, können Sie sich davon überzeugen, zu welch höflichem Umgangston Ihr VC-20 in der Lage ist!

Einführung in die Datenfelder (Variable)

Viele der in diesem Handbuch benutzten Programme verwenden Datenfelder (Variable), die der Vereinfachung und Verbesserung dienen.

Variable sind sehr nützlich, da sie im Zusammenhang mit Zahlen, Formeln, graphischen Symbolen, Wörtern, Textpassagen oder auch ganzen Sätzen eingesetzt werden können. Beispiele für die Namen dieser Variablen sind: X, AB, S2, X$ (sprich X-String), AB$, S2$. Um Ihnen eine Vorstellung zu geben, wie leistungsfähig Variable sind, zeigt Ihnen die Tatsache daß Variable in der Lage sind, bis zu 255 Zeichen zu repräsentieren.

Es stehen 2 Arten von Datenfeldern zur Verfügung: das numerische Datenfeld und das alpha-numerische Datenfeld. Das numerische Datenfeld wird dazu verwendet, Zahlen zu speichern. Das alpha-numerische Datenfeld ist in der Lage, alle möglichen Arten von Zeichen aufzubewahren (Zahlen, Buchstaben, graphische Zeichen, Steuerzeichen für Farben und Cursor, usw.)



Variable könnte man auch als Ablageschränke eines Computer-Systems sehen. Dabei ist es notwendig, eine Aufteilung rein numerischer und alpha-numerischer Werte zu machen, die dem VC-20 durch spezielle Namen mitgeteilt werden. Die Namen von numerischen Variablen können 1 oder 2 Stellen lang sein und bestehen aus einem oder zwei Buchstaben oder einem Buchstaben und einer Ziffer. Es folgen einige Beispiele:

X   AB   S2   C2   AA   ZX

Die Namen für alpha-numerische Variable können 1 bis 3 Zeichen lang sein (einschl. des $-Zeichens), müssen jedoch immer mit einem Buchstaben (A-Z) beginnen und durch ein Dollarzeichen ($) beendet werden. Dazu einige Beispiele:

X$ AB$ S2$ C2$ AA$ ZX$

In keinem Fall darf die Variable mit einer Zahl beginnen. Dies könnte der Computer als Zeilennummer interpretieren.

Es folgt ein Kurzprogramm, welches den Einsatz von Variablen demonstriert:

10 A$ = „VC-20“

20 PRINT „HALLO,“ A$

RUN

Der VC-20 wird den Text „HALLO, VC-20“ auf den Bildschirm bringen. Warum? Weil in der Zeile 10 definiert wurde, daß das Datenfeld A$ den Text „VC-20“ enthält. Probieren Sie nun:

10A = 2

20B = 3

30C = 4

40 PRINT A*B*C

In diesem Beispiel ist A*B*C gleichbedeutend mit 2*3*4, da durch die Eingaben der Zeilen 10-30 die Werte 2, 3 und 4 in die Datenfelder A, B und C gespeichert wurden.

Hier ein weiteres Beispiel, welches zwei Arten von Variablen in Verbindung mit der INPUT-Anweisung verwendet. Die INPUT-Anweisung erlaubt es Ihnen, den Inhalt der Variablen zu bestimmen:

10 PRINT „WELCHES WORT SOLL IN X$ STEHEN“: INPUT X$

20 PRINT „WELCHER WERT SOLL FELD X BEINHALTEN“: INPUT X

30 PRINT „X$ IST JETZT GLEICH“ X$

40 PRINT „X IST JETZT GLEICH“ X

RUN

Auswahl einer Note

Für dieses Eingabebeispiel soll jetzt wieder einmal mit der Tonerzeugungs-Fähigkeit des VC-20 experimentiert werden. Eingabe von:

```
NEW
10 INPUT „NOTENHOEHE"; H
10 IF H = 0 THEN 90
30 INPUT „LAENGE DER NOTE"; L
40 POKE 36878,15
50 POKE 36875,H
60 FOR T = 1 TO L:NEXT
70 POKE 36878,0
80 GOTO 10
90 END
```


Nach Betätigung der RETURN -Taste sieht der Bildschirm wie folgt aus:

```
NOTENHOEHE? 225

LAENGE DER NOTE? 1000

RUN
```


Nach Betätigung der RETURN -Taste ertönt eine bestimmte Note mit einer bestimmten Länge. Nach jeder Note ist es möglich eine andere zu generieren, womit Sie Erfahrung im übersetzen von POKE-Befehlen in Noten sammeln können.

Die erste Zeile des Beispiels verschafft uns die notwendige Flexibilität, die Höhe der Note in den Zeilen 50 und 60 zu steuern. Über die Zeile 20 wird bestimmt, wann das Programm zum Ende gebracht wird (Eingabe des Wertes 0 bei Frage NOTENHÖHE?). Die Zeile 70 schaltet den Ton aus, der in Zeile 40 aktiviert wurde.

Die wichtigsten Details im Zusammenhang mit der Erzeugung von Musik befinden sich im Kapitel 5.

Die GET-Anweisung

Jetzt, da Sie sich mit der INPUT-Anweisung vertraut gemacht haben, können wir uns mit einer eleganteren Lösung für die Sammlung von Informationen und Daten über die Tastatur befassen.

Die GET-Anweisung wird dazu verwendet, die über Tastatur eingegebenen Zeichen Stück für Stück in den Speicher zu bringen. Es ist in der Tat so, daß Sie noch nicht einmal die RETURN -Taste betätigen müssen! Und so sieht das aus:

10 GET A$

Wie ist es nun möglich, ein Programm zum Anhalten zu bringen und darauf zu warten, daß etwas eingegeben wird? Dies erreichen wir durch eine Schleife, in der das Programm ständig prüft, ob eine Taste gedrückt wird (IF . . . THEN).

10 GET A$

20 IF A$ = "" THEN 10

Kein Leerzeichen zwischen den Anführungszeichen

Was ist der Sinn dieser Anweisung? Als sehr einfaches Beispiel dient das obige, 2zeilige Programm, welches so lange wartet, bis eine beliebige Taste gedrückt wurde. Dies ist dann sehr praktisch, wenn Informationen auf dem Bildschirm festgehalten werden sollen, bis diese jemand gelesen hat um dann (durch Drücken einer beliebigen Taste) im Programm fortzufahren.

Hier eine erweiterte Version der GET-Anweisung:

10 GET A$

20 IF A$ = "" THEN 10

30 IF A$ = "H" THEN PRINT "HUEHNERSUPPE"

40 IF A$ = "S" THEN PRINT "SPAGHETTI"

50 IF A$ = "K" THEN PRINT "KOTELETTE"

60 GOTO 10

RUN

Wurde das Programm durch RUN zum Laufen gebracht, dann wartet es bis eine Taste gedrückt wird. War die Taste allerdings das H, erscheint die Nachricht HUEHNERSUPPE, bei S die Nachricht SPAGHETTI und bei K die Nachricht KOTELETTE auf dem Bildschirm. Durch die Eingabe eines einzelnen Buchstabens ist der VC-20 also in der Lage, mit ganzen Wörtern zu antworten.

Es folgt das bisher längste Programm, ein praktisches Beispiel der GET- und PRINT-Anweisungen – ein *Kochrezept-Programm.* Sie sollten sich nicht vor der Eingabe dieses Programms wegen dessen Größe scheuen. In den meisten Fällen werden sehr einfache PRINT-Anweisungen verwendet. Hierbei soll lediglich die Benutzung der Variablen A$ demonstriert werden, die für die Eingabe ganzer Sätze eingesetzt wird, und wie die GET-Anweisung praktisch zum Einsatz kommen kann.

```
10 PRINT " [SHIFT] [CLR HOME] BITTE WÄHLEN SIE";
20 PRINT "AUS DEM SPEISEPLAN"
30 PRINT
40 PRINT "H . . . HUEHNERSUPPE"
50 PRINT "S . . . SPAGHETTI"
60 PRINT "K . . . KOTELETTE MIT EI"
200 GET A$: IFA$ = "" THEN 200
210 IF A$ = "H" THEN 500
220 IF A$ = "S" THEN 700
230 IF A$ = "K" THEN 900
240 GOTO 200
500 PRINT " [SHIFT] [CLR HOME] HUEHNERSUPPE SPEZIAL"
510 PRINT
520 PRINT "MAN NEHME EIN HUHN"
530 PRINT "RUPFE ES UND ENTFERNE"
540 PRINT "INNEREIEN"
545 PRINT "KOCHE 2 LTR WASSER"
550 PRINT "IN GROSSEM TOPF"
560 PRINT "HUHN HINEINLEGEN"
570 PRINT "UND 2 STD. KOCHEN"
580 PRINT "ODER BIS HUNGERGEFÜHL"
585 PRINT "UNERTRÄGLICH WIRD"
590 PRINT
```

Selbstverständlich können Sie auch Ihre eigenen Kochrezepte eingeben. Es sind lediglich die Überschriften und Zutaten zu ändern.

Das bedeutet, daß dieses Programm so lange wartet, bis entweder H, S oder K eingegeben wird. Andere Eingaben werden ignoriert.

Für eine bessere Lesbarkeit sind einige Sätze auf mehrere Zeilen verteilt.

Die Eingabe einer PRINT-Anweisung ohne Inhalt bringt eine Leerzeile auf den Bildschirm.

```
600 PRINT "DRUECKE EINE"
605 PRINT "BELIEBIGE TASTE"
607 PRINT "ZUR FORTSETZUNG"
610 GET A$:IF A$ = "" THEN 610
620 GOTO 10
700 PRINT " [SHIFT] [CLR HOME] SPAGHETTI ROMA"
710 PRINT
720 PRINT "1 PFUND HACKFLEISCH"
730 PRINT "MIT 1 ZWIEBEL ANBRATEN"
740 PRINT "1 GRUENE PAPRIKA"
750 PRINT "DAZU 1 DOSE TOMATEN"
760 PRINT "200 GR. TOMATENMARK"
770 PRINT "0,2 LTR. WASSER"
780 PRINT "3 KNOBLAUCHZEHEN, SALZ"
790 PRINT "UND PFEFFER,"
795 PRINT "ROT. PFEFFER & OREGANO"
800 PRINT "1 STD. SIEDEN UND MIT"
810 PRINT "SPAGHETTI SERVIEREN"
820 GOTO 590
900 PRINT " [SHIFT] [CLR HOME] KOTELETTE MIT EI"
910 PRINT
920 PRINT "MAN BRATE 1 KOTELETTE"
930 PRINT "EBENSO 1 EI"
940 PRINT "UND REICHE DAZU SALAT"
950 PRINT "UND POMMES FRITES"
960 GOTO 590
```

Wurde das Programm fehlerlos eingegeben und RUN eingegeben, dann sollte folgendes auf dem Bildschirm erscheinen:

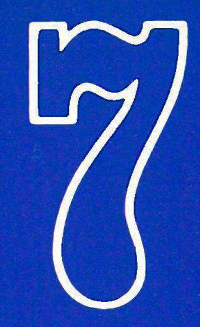
```
BITTE WAEHLEN SIE
AUS DEM SPEISEPLAN:

H... HUEHNERSUPPE
S... SPAGHETTI
K... KOTELETTE MIT EI
```

Nun wartet der VC-20 darauf, daß der Buchstabe H, S oder K gedrückt wird. Bei Betätigung einer jeden anderen Taste tut sich gar nichts (die Zeilen 200 bis 240 sind dafür verantwortlich). Bei Drücken von „H", erscheint das Rezept für „Hühnersuppe" auf dem Bildschirm, Schadensersatzanforderungen werden allerdings nicht entgegengenommen.

Die Rezepte können selbstverständlich ganz nach Lust und Laune verändert und erweitert werden. Um ein weiteres Rezept diesem Kochbuch hinzuzufügen, ist lediglich nach der Zeile 60 eine PRINT-Anweisung einzuschieben und eine IF-Anweisung nach der Zeile 230. Das Kochrezept selbst kann einer beliebigen Stelle am Ende des Programms hinzugefügt werden. Die letzte Zeile des Programms sollte selbstverständlich den Befehl GOTO 590 beinhalten, was für die Programmfortsetzung bedeutet, daß eine Taste gedrückt werden muß.

Dieses hier beschriebene Programm kann auch für andere Zwecke als die eines Kochbuches eingesetzt werden. So z. B. ist es möglich, eine Adressendatei aufzubauen. Anstatt der Menüs könnten zuerst die Namen mit entsprechenden Initialen erscheinen. Anstatt der eigentlichen Kochrezepte würden dann an diesen Stellen die Namenadressen und Telefonnummern der Personen aufgeführt. Wir schlagen vor, daß Sie sich einmal Ihre eigenen Gedanken darüber machen, wie die GET- und INPUT-Anweisungen im Zusammenhang mit Programmen auch anderweitig eingesetzt werden können.



Einführung in die Programmierung

- Die ersten BASIC-Befehle
- Zufallszahlen

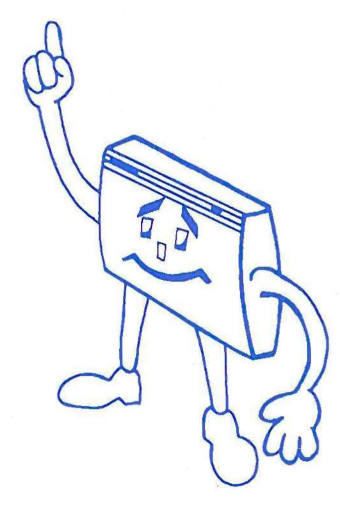
Programmvorschlag

Schreiben Sie folgendes Programm genau und sehen Sie was passiert:



Halten Sie das Programm mit: 

an.



Die ersten BASIC-Befehle

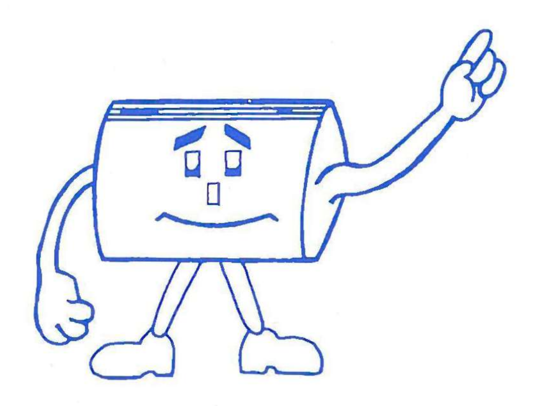
Bis jetzt waren Sie noch geduldig und haben mehrere Programme geschrieben, ohne (möglicherweise) auf Anhieb zu verstehen, wie sie arbeiten. Dieses Kapitel erklärt die Einzelkomponenten dieser trickreichen kleinen Programme, mit dem Ziel, Ihnen einen Leitfaden auf den Weg zur Programmierung des VC-20 zu geben.

PROGRAMM 1: IHR NAME IM RAMPENLICHT
(Kapitel 2)

```
10 PRINT " [SHIFT] [CLR HOME]
20 FOR T = 1 TO 300: NEXT
30 PRINT "Ihr Name"
40 FOR T = 1 TO 300: NEXT
50 GOTO 10
```

Dieses Programm und alle in diesem Kapitel folgenden Programme sind den Beispielprogrammen entnommen, die am Anfang eines jeden Kapitels aufgeführt sind.

Oberflächlich betrachtet scheint dieses Programm nicht besonders leistungsfähig zu sein. Versteht der Benutzer jedoch erst einmal die Einzelteile, dann hat er damit den Schlüssel zur kreativen Entwicklung der Zeichtrick-Darstellung. Die Zeile 10 ist eine PRINT-Anweisung (Ausgabe auf Bildschirm) mit einem Zeichen innerhalb der Anführungsstriche das den Bildschirm löscht. Würde versucht werden, diese Eingabe ohne Verwendung der Anführungsstriche einzugeben, dann würde diese Zeile verschwinden, noch bevor sie vollständig ist. *Der VC-20 erkennt eine Zeile als neu eingegeben an, wenn der Cursor sich auf dieser Zeile befindet, und RETURN gedrückt wird.*



VC-20 TIP:

Was geschieht, wenn die Anführungszeichen-Taste gedrückt wird? Beim erstmaligen Drücken passiert etwas außergewöhnliches. Die danach folgende Betätigung von HOME oder CLR oder auch der Cursor-Steuertasten (hoch, runter, rechts, links) erzeugt ein *graphisches Zeichen* in negativer Darstellung. Das passiert auch dann, wenn die Farb-Steuerzeichen gedrückt werden. D. h., daß innerhalb von Anführungszeichen Steuertasten-Informationen als Programmierbefehle erkannt werden. Daher zeigt der VC-20 ein Sonderzeichen nach der Betätigung einer Farb- oder Cursor-Steuertaste an, um damit die jeweils relevante Operation zu kennzeichnen. So z. B. wird ein Herz in negativer Bildschirmdarstellung gezeigt, wenn folgende Eingabe erfolgt:

"und SHIFT CLR HOME "

Innerhalb eines Programms verursachen diese Eingaben die Löschung des gesamten Bildschirms. Selbstverständlich gibt es noch andere Zeichen, die im Zusammenhang mit anderen Operationen zu sehen sind. Wird das Anführungszeichen zum *zweiten* Mal eingegeben, dann arbeiten die Farb- und Cursor-Steuertasten wieder im Normal-Modus.

Die INSERT-Taste hat eine ähnliche Wirkung. Wenn diese Taste gedrückt wird, dann benimmt sich jedes Leerzeichen dieser Zeile so, als ob es sich innerhalb von Anführungszeichen befände. Dasselbe gilt auch für die Cursor-Steuerzeichen. Ferner wird durch das Drücken der DELETE-Taste (Löschung) ein graphisches Zeichen im negativen Bildschirm-Darstellungsformat in diese Positionen gebracht, die dann Zeichen bei der Auflistung des Programms (LIST) löschen und DELETEs auf dem Bildschirm aufgeben, wenn PRINT aktiviert wird.

Bei der Zeile 20 dieses Programms handelt es sich um eine *Zeitverzögerungs-Schleife*. Dabei befinden sich eigentlich auf der gleichen Zeile zwei BASIC-Anweisungen, die durch einen Doppelpunkt (:) voneinander getrennt sind. Es geschieht weiter nichts, als daß der VC-20 von 1 bis 300 zählt. Dies dient lediglich der Verlangsamung des Programmablaufs. (Löschen Sie doch mal die Zeilen 20 und 40. Sehen Sie, es blinkt zu schnell.)

In der Zeile 30 haben Sie Ihren Namen eingegeben (hoffen wir zumindest). Dieser Befehl verursacht die Ausgabe des Namens auf dem Bildschirm. Es befindet sich keine andere Information auf dem Bildschirm, da dieser schon durch den Befehl der Zeile 10 gelöscht wurde.

In der Zeile 40 befindet sich wiederum eine Zeitverzögerungs-Schleife, damit Ihr Name so lange auf dem Bildschirm verbleibt, um gelesen werden zu können.

Der Befehl der Zeile 50 verursacht den Sprung zur Zeile 10, wo der Bildschirm wieder gelöscht wird.

Damit befindet sich das Programm in einer Programmschleife die dazu führt, daß Ihr Name kurz auf dem Bildschirm erscheint, gelöscht wird, um anschließend wieder dort zu erscheinen. Somit gibt es für das Auge den Anschein, als ob der Name blinkt.

Sie sollten jetzt etwas mit diesem Programm experimentieren. So z. B. können die Zeitintervalle für den Blinkrhythmus verändert werden, indem der Wert 300 der Zeilen 20 und 40 durch einen neuen ersetzt wird.

PROGRAMM 2: VIELE HERZCHEN
(KAPITEL 3)

```
10 FOR H = 1 TO 505
20 PRINT " ♥ ";
30 NEXT
40 FOR C = 8 TO 255 STEP 17
50 POKE 36879, C
60 FOR T = 1 TO 500: NEXT
70 NEXT
80 GOTO 40
```

Dieses Programm unterhält Sie mit farbenprächtigen Herzchen auf dem Bildschirm. Es kann Ihnen aber auch die Bedeutung von Satzzeichen innerhalb eines Programms deutlich machen, ebenso das Wechseln der Bildschirm-Farben durch POKE.

Bei der Zeile 10 handelt es sich um eine Programmschleife mit einem Zähler, der von 1 bis 505 zählt. Damit werden 505 Herzchen auf den Bildschirm gedruckt (PRINT). Hätten Sie versucht, auch den 506. Bildschirmplatz mit einem Herz zu belegen, hätte dies den automatischen Zeilenvorschub in Gang gesetzt, welche die oberste Zeile verschwinden läßt, um auf der untersten Zeile *Platz zu schaffen.* Letztlich wären dadurch *weniger* Herzchen auf dem Bildschirm, als es mit dem hier aufgeführten Programm möglich ist.

Die Zeile 20 bringt die Herzchen durch die PRINT-Anweisung auf den Bildschirm. Dabei hat der am Ende dieser Zeile stehende Strichpunkt (;) eine ganz besondere Aufgabe. Bei einem PRINT-Befehl ohne ";" wird der Cursor zum Anfang der neuen Zeile gebracht (wie bei einem *Wagenrücklauf* bei einer Schreibmaschine). Der Strichpunkt jedoch weist den VC-20 an, daß diese *Wagenrücklauf-Aktion* nicht zu erfolgen hat. Dadurch wird erreicht, daß die nächste Plazierung des Herzchens direkt dem vorhergehenden erfolgt.

Die Zeile 30 vervollständigt diese Programmschleife. So lange die „Laufvariable H" nicht den Wert 505 erreicht hat, werden Herzchen ausgegeben. Nach dem letzten Herzchen (505) fährt das Programm mit dem Befehl der nächsten Zeile fort (Zeile 40).

Mit Zeile 40 beginnt eine neue Programmschleife die den Befehl der Zeile 50 beinhaltet. Sie ist für den Farbenwechsel von Hintergrund und Rahmen verantwortlich. „C" ist die Laufvariable, die ausgehend vom Wert 8 in Schritten von „17" bis 255 erhöht wird. Jedes Mal, wenn die Zeile 70 (NEXT) erreicht wurde, ist die Variable „C" um 17 erhöht worden. Dadurch wird ein Farbenzyklus aufgestellt, einschl. schwarz für Hintergrund und Rahmen, und weiß auch wiederum für Hintergrund und Rahmen, usw. (Abhandlung aller 8 Farben für Bildschirm-Rahmen). Danach sehen Sie noch die weiter möglichen Farben für den Bildschirmhintergrund (ohne Änderung des Rahmens). (Siehe Anhang I – Auflistung der Farbselektions-Nummern.)

Der Befehl der Zeile 50 nimmt die Farbänderung vor. Der Wert im Feld C wird in die Adresse 36879 gePOKEt.

Zeile 60 enthält eine Zeitverzögerungs-Schleife. Bei deren Fehlen könnten Sie durch den raschen Farbwechsel mit Kopfschmerzen rechnen.

Zeile 70 schließt die Programmschleife, die mit Zeile 40 anfängt. Dieser Befehl hätte auch „NEXT C" lauten können.

Zeile 80 verursacht den Rücksprung des Programms auf Zeile 40, womit der Farbwechsel wieder von neuem beginnt. Das Programm wird so lange laufen, bis die STOP-Taste gedrückt ist oder der VC-20 ausgeschaltet wird. Die Eingabe des Befehls POKE 36879, 27 nach der Betätigung der STOP-Taste wird den Bildschirm wieder in den Normalfarben-Status zurückversetzen.

PROGRAMM 3: TURNENDES VC-20-MÄNNCHEN
(KAPITEL 4)

```
10 PRINT "{CLR HOME}";
20 PRINT "⧅⧇⧄"
30 PRINT "□▩□"
40 PRINT "⧄□⧅"
50 FOR T=1 TO 300: NEXT
60 PRINT "{CLR HOME}";
70 PRINT "□⧇□"
80 PRINT "⧄▩⧅"
90 PRINT "□◧◧"
100 FOR T=1 TO 300: NEXT
110 GOTO 10
```

Dieses Programm ist dem sehr ähnlich, welches *Ihren Namen blinken ließ.* Anstatt den Namen auszugeben und diesen nach kurzer Zeit zu löschen, wie dies vom ersten Programm ausgeführt wurde, zeigt dieses Programm ein fertiges Bild, pausiert, und ersetzt dies durch ein neues. Kopf und Körper des VC-20-Männchens verbleiben in der gleichen Position, währenddem sich Arme und Beine zu bewegen scheinen.

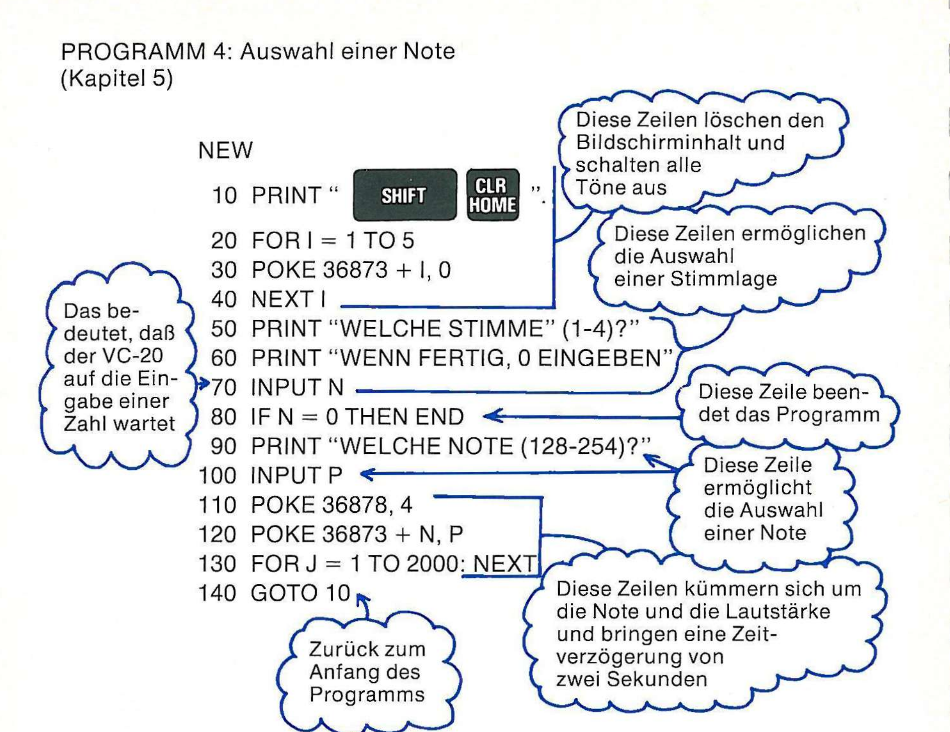
Zeilen 10 und 60 bringen den Cursor in die oberste linke Ecke des Bildschirms (HOME). Dies ermöglicht das Erscheinen des Männchens immer auf der gleichen Bildschirmposition.

Die Zeilen 20, 30 und 40 „zeichnen" die anfängliche Haltung des Männchens.

Die Zeilen 50 und 100 sind Zeitverzögerungs-Schleifen, die bewirken, daß das Bild lange genug auf dem Bildschirm verbleibt.

Zeilen 70, 80 und 90 „zeichnen" die zweite Haltung des Männchens, aber so schnell, daß die Bewegung vom Auge als „sofort" registriert wird.

PROGRAMM 4: Auswahl einer Note
(Kapitel 5)



Nach der Überprüfung aller Zeilen kann jetzt das Programm ausgeführt werden (RUN und RETURN). Das Programm ermöglicht es, eine Stimmlage und eine Note auszuwählen und diese Note für etwa 2 Sekunden zu spielen. Danach schaltet sich der Ton automatisch aus, und das Programm bittet um die Eingabe einer weiteren Stimmlage und Note. Dieses Programm ist für die musikalisch experimentierenden Benutzer bedeutsam und sollte aus diesem Grund nicht ausgelassen werden. Zur Beendigung des Programms ist bei der Stimmlage eine „0" einzugeben. Bei der Eingabe von Stimmlage 3 und Note 254 gibt die Anlage keinen Ton von sich. Hierbei handelt es sich nicht um einen Fehler – diese Note ist einfach zu hoch für das menschliche Ohr. (Wenn Sie jedoch Ihren Hund als „Versuchskaninchen" einsetzen wollen, sollte dieser entsprechend reagieren.)

PROGRAMM 5: Umrechnung der Temperatur-Skalenwerte (Kapitel 6)

```
10 INPUT "GRAD FAHRENHEIT"; F

20 PRINT F "GRAD F."

30 PRINT "IST" (F-32)*5/9 "GRAD C."

40 PRINT

50 GOTO 10
```

Dieses Programm führt Sie in die INPUT-Anweisung ein. „INPUT" hält die Verarbeitung des Programmes an und macht sich durch ein Frage-Zeichen bemerkbar. Geben Sie hier die geforderten Eingaben ein.

Zeile 10 richtet an Sie die Frage GRAD FAHRENHEIT? auf dem Bildschirm. Die sich innerhalb der Anführungszeichen einer INPUT-Anweisung befindenden Wörter sind gleichzusetzen mit der PRINT-Anweisung. Jedoch folgt auf den entsprechenden Ausdruck immer ein Fragezeichen, wobei das Programm auf Antwort wartet.

Die Zeile 20 wiederholt Ihre Eingabe des Fahrenheit-Wertes. Die Zeile 30 stellt das Umrechnungsergebnis dar. Die Zeile 40 enthält eine PRINT-Anweisung, ohne gleichzeitig eine Nachricht auszugeben. Damit wird auf dem Bildschirm eine Leerzeile erzeugt.

Letztlich führt der Befehl der Zeile 50 zum Programmanfang zurück, wonach noch mehr Daten von Ihnen angefordert werden. Die ursprüngliche Frage wird wieder gestellt, und sie können weitere Fahrenheit-Werte zur Konvertierung eingeben. Soll der Programmablauf jedoch angehalten werden, dann ist die STOP-Taste zu halten und die RESTORE-Taste zu drücken. Es gibt keine andere Möglichkeit, das Programm in seinem Ablauf anzuhalten.

PROGRAMM 6: Pseudo-Irrgarten
(Kapitel 7)

```
10 PRINT " [SHIFT] [CLR HOME] "
20 PRINT CHR$(205.5 + RND(1));
30 GOTO 20
```

Dieses Programm erzeugt einen Pseudo-Irrgarten auf dem Bildschirm. Wie Sie vermuten, befindet sich des Rätsels Lösung in Zeile 20.

Die CHR$-Funktion ergibt ein Zeichen, welches auf einer Zahl von 0-255 basiert. Jedes von dem VC-20 auf den Bildschirm gebrachte Zeichen ist auf diese Art und Weise verschlüsselt (siehe Anhang H). Um die Kodierung eines bestimmten Zeichens zu erfahren, braucht lediglich der Befehl

PRINT ASC ("X")

eingegeben zu werden, wobei "X" das Zeichen darstellt, von welchem wir die (ASCII-)Kodezahl erfahren wollen. Wenn Sie nun

PRINT CHR$ (X)

eingeben, wobei "X" die Zahl ist, die Sie soeben ermittelt haben, erfahren Sie wieder das ursprüngliche Zeichen. Verstehen Sie jetzt wie es funktioniert?

Wenn Sie jetzt PRINT CHR$ (205); CHR$ (206) eingeben, erscheinen die zwei, sich auf der rechten Seite der „M"- und „N"-Tasten befindenden graphischen Zeichen auf dem Bildschirm. Dieses sind die zwei Zeichen, die für die Erstellung des Pseudo-Irrgartens verwendet werden.

Durch die Eingabe der Formel 205.5 + RND(1) wird der VC-20 eine zufällige Zahl errechnen, die zwischen 205.5 und 206.5 liegt. Es besteht also eine 50%ige Möglichkeit, daß die ausgewählte Nummer über oder unter 206 liegt. Die CHR$-Funktion nimmt aber nur den ganzzahligen Teil einer Zahl. Aus diesem Grund wird die Hälfte der Zeichen aus der Kodierung 205 und die andere Hälfte aus der Kodierung 206 zusammengesetzt.

Einige Experimente mit diesem Beispiel können darin bestehen, daß dem Wert 205.5 einige Zehntel-Werte hinzuaddiert oder von diesem subtrahiert werden. Dies würde die bisherige Chancengleichheit der beiden Zeichen in die eine oder die andere Richtung verändern.

Zufallszahlen

Die Funktion zur Erzeugung zufälliger Zahlen ist die nützliche Einrichtung der BASIC-Sprache, wenn es darum geht, auf Zufall beruhende Programme zu entwickeln.

Die Eingabe X = RND(1) wird den VC-20 veranlassen, eine Zahl zwischen 0 und 1 (ausschließlich) auszuwählen und diese in das Feld X zu setzen. Daraus resultiert eine Bandbreite möglicher Werte für X:

$0 < X < 1$

Wenn Sie mit Zufallszahlen arbeiten, dann denken Sie daran, daß Sie durch Berechnung die Bandbreite variieren können. Sollte es z. B. erwünscht sein, einen Satz möglicher Werte zwischen 0 und 3 zu erhalten, dann könnte ganz einfach X mit dem Wert 3 multipliziert werden. Die neue Bandbreite ist wie folgt:

$0 < X < 3$

Wie sollte die Berechnung zur Änderung einer Bandbreite aussehen, wenn eine Zahl zwischen 10 und 20 zu selektieren ist? Die Addition der Zufallszahl mit 10 ergibt

$10 < X < 11$

Durch die Multiplikation der Zufallszahl mit dem Faktor von 10 vor der Addition von 10 wird die Bandbreite auf folgendes gesetzt:

$10 < X < 20$

Die Formel für Zufallszahlen zwischen 10 bis 20 ist wie folgt:

X = RND(1)*10 + 10

Bis jetzt wurde besprochen, wie sich die Bandbreite von Zufallszahlen ändern läßt. Jedoch enthalten die Ergebnisse Zahlen mit Dezimalstellen, die in den meisten Fällen, wie z. B. beim Würfelspiel unerwünscht sind. Der INT-Befehl kann als bereinigende Funktion eingesetzt werden. Alle Dezimalstellen werden von der ausgewählten Zahl abgetrennt. Die Formel für eine Zufallszahl zwischen 10 und 20 anhand der INT-Funktion ist wie folgt:

X = INT(RND(1)*10 + 10)

Die Bandbreite der möglichen Zahlen ist wie folgt:

10 < = X < = 19

Aber einen Augenblick mal! Der obere Schwellenwert der Bandbreite ist von 20 auf 19 gesenkt worden. Warum? Weil vorher das Bandbreitenende immer kleiner als 20 war. Die INT-Funktion wird alle Dezimalstellen von einer Zahl größer als 19 und kleiner als 20 abtrennen, was in 19 resultiert. Am anderen Ende der Bandbreite werden alle zwischen 10 und 11 liegenden Werte auf 10 geändert. Sollte dennoch die eigentliche Bandbreite zwischen 10 und 20 liegen, dann sollte die Formel folgendermaßen lauten:

X = INT(RND(1)*11 + 10)

Durch multiplizieren erweitern Sie die Bandbreite, durch Addition verschieben Sie diese in andere Zahlenbereiche.

Die allgemeine Formel zum Aufsetzen von willkürlichen Zahlen innerhalb einer bestimmten Bandbreite ist wie folgt:

X = INT(RND(1)*a) + b

Dabei entspricht „a" der Anzahl der möglichen Werte und „b" dem niedrigsten Wert der Bandbreite.

Anhang

A: ● Zusatzeinrichtungen
- ● Der Heim-Computer VC-20
- ● Anschließbare Module und Geräte
- ● Software

B: ● Arbeiten mit Bandkassette

C: ● VC-20 BASIC
1. Daten und Operatoren
2. Befehle
3. Anweisungen
4. Funktionen

D: ● Abkürzungen der BASIC-Schlüsselwörter

E: ● Farbkombinationen des Bildschirm-Hintergrunds und -Rahmens

F: ● Tabelle der Musiknoten

G: ● 20 Sondereffekte des VC-20

H: ● Kodierung des Zeichenvorrats

I: ● Hauptspeicheraufteilung für den Bildschirm

J: ● ASCII- und CHR$-Kode

K: ● Ableitung mathematischer Funktionen

L: ● Anschlüsse für Eingabe-/Ausgabe-Geräte

M: ● Probeprogramm für den VC-20

N: ● Fehlermeldungen

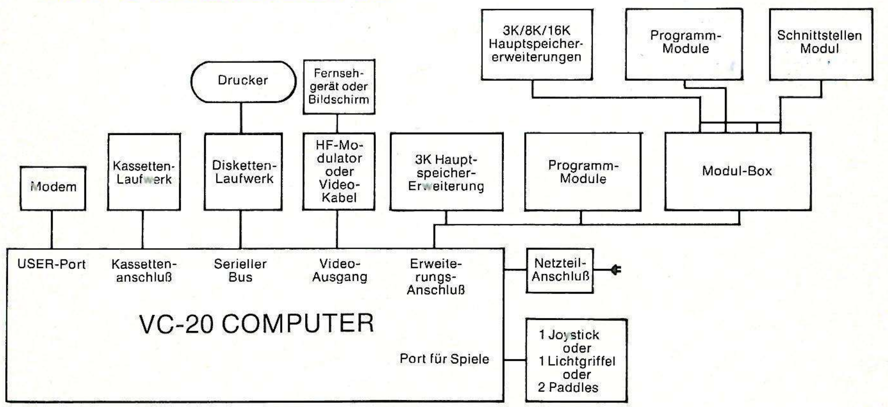
ANHANG A: ZUSATZEINRICHTUNGEN

Ein kurze Einführung

Dieses Handbuch ist für den Personenkreis gedacht, der sich zum ersten Mal mit Computer-Anlagen auseinandersetzt. Aus diesem Grund beabsichtigen wir an dieser Stelle nicht, der Erklärung von zusätzlich erhältlichen Geräten, die sich an den VC-20 anschließen lassen, einen großen Platz einzuräumen. Für jedes einzelne Zusatzgerät für den VC-20 ist ein separates Handbuch erhältlich, welches erklärt, wie das Gerät anzuschließen ist und wie es für sehr interessante Aktivitäten eingesetzt werden kann.

Der VC-20 wurde so konzipiert, daß er mit den Anforderungen des Benutzers wachsen kann, wobei von COMMODORE ein allgemeiner Plan zur Verfügung gestellt werden kann, der die einzelnen Zusatzgeräte und deren Zusammenwirken im Zusammenhang mit dem VC-20 erklärt. Hier eine Übersicht zu diesem Plan.

Der Heim-Computer VC-20 von COMMODORE*



Anschließbare Module und Geräte

Super-Erweiterungs-Modul
- 3 KB Zusatz-Hauptspeicher (Erweiterung der VC-20-Kapazität auf 8 KB)
- Graphik- und Plotting-Befehle mit hohem Auflösungsvermögen (high resolution = über 31000 Bildpunkte)
- Vorherig zugeordnete Funktionstasten
- Tastatur als Klaviatur (jede Taste 1 Ton)

* Die Lieferbereitschaft der hier beschriebenen Zusatzgeräte ist innerhalb 1981 geplant.

Programmierhilfe-Modul
- Hilfswerkzeug für Programmierer („Tool-Kit")
- Maschinensprache-Monitor
- Vorherig zugeordnete Funktionstasten (Programmierungsbefehle)
- Vom Anwender zuordnungsbare Funktionstasten

Kassetten-Rekorder

Das Zusatzgerät von größtem Interesse ist der Bandkassetten-Rekorder von COMMODORE. Dieses Kassettengerät ist in der Lage, mehrere Tausend Zeichen (Buchstaben und Nummern) auf einer normalen Bandkassette zu speichern. Damit ist der Benutzer im Stande, eine ganze Reihe großer Programme auf das Band laden und diese schnell wieder abrufen zu können. Es entfällt somit die Neueingabe des Programms, jedes Mal, wenn dieses Programm auszuführen ist. (Siehe Anhang B: Arbeiten mit Bandkassetten).

Das VC-20-Bandkassetten-Gerät wird mit einem Spezialstecker an der Kassetten-Schnittstelle angeschlossen, die sich an der Rückseite des VC-20 befindet. Eine weitere Schnittstelle wird nicht benötigt. COMMODORE stellt eine Vielfalt von Computer-Programmen auf Band zur Verfügung, die über den Kassetten-Rekorder eingelesen werden können.

Modul-Box

Der Benutzer ist in der Lage, die Modul-Box mit dem Anschluß an der Rückseite des VC-20 zu verbinden. Dieses Gerät erlaubt es, zur gleichen Zeit mehr als ein Modul einzusetzen. Sie ist mit 6 Anschlüssen versehen und überwacht und steuert die Verbindung zwischen dem VC-20 und den Modulen und Hauptspeicher-Erweiterungen.

Diskettenlaufwerk

Das VC-Diskettenlaufwerk kann bis zu 170.000 Zeichen auf einer Diskette (Magnetscheibe) speichern. Das Laden und Speichern von sehr langen Programmen geschieht innerhalb weniger Sekunden. Dabei werden u. U. längere Wartezeichen ausgeschaltet, die bei der Verwendung eines Bandkassetten-Geräts relativ häufig vorkommen. Das Gerät wird an den seriellen Bus des VC-20 angeschlossen.

IEEE-488-Schnittstellen-Modul

IEEE-488 ist eine universelle technisch/wissenschaftliche Standard-Schnittstelle die in der Lage ist, von COMMODORE hergestellte Peripherie-Einheiten, wie z. B. Disketten-Laufwerke und Drucker sowie technisch/wissenschaftliche Instrumente und Werkzeuge zum Anschluß an den VC-20 zu überwachen und zu steuern.

Serieller Drucker

Der VC-20 verfügt über eine serielle Schnittstelle für Peripherie-Geräte, die serielle Verbindung zur Kommunikation mit der Computer-Anlage verwenden. Dieser serielle Anschluß ermöglicht es dem Benutzer, seine Programmlisten und Ergebnisse, mittels des COMMODORE Matrix-Druckers (VC 1515), auf Papier zu bringen.

Anschlußport für Spiele

Dieser Spezialanschluß für Spiele erlaubt den Anschluß von sogenannten Joysticks (Steuerknüppel) und Lichtgriffeln, so daß es für den Benutzer möglich ist, aufregende Bildschirm-Spiele durchzuführen, ohne dabei die Tastatur des VC-20 einzusetzen.

Daten-Fernübertragung

Vielleicht haben Sie schon von Programmen gelesen oder gehört, die es ermöglichen, daß sich Ihr eigener Heim-Computer mit einem anderen System „unterhält". Möglicherweise erhält man dadurch Informationen wie z. B. Börsennotierungen, Informationen der Wirtschaft, Nachrichten und andere Daten die von Interesse sind. Der VC-20 ist schon mit dieser Eigenschaft ausgerüstet.

Der Anschluß des VC-20 ist mit einer V24-Schnittstelle ausgerüstet die es ermöglicht, unter Verwendung eines Modems, Informationen über Telefonleitungen auszutauschen.

Software

Zusätzlich zu der hier aufgeführten Hardware steht dem Benutzer noch eine ganze Palette von Software zur Verfügung.

Einige dieser Programme sind auf Programm-Moduln gespeichert, die unmittelbar an den VC-20 angeschlossen werden können und funktionsfähig sind, sobald die Anlage eingeschaltet ist. Einige sind auf Band und andere wiederum auf Diskette gespeichert, und viele davon stehen in dem Unterrichtshandbuch zum VC-20, welches zur programmierten Selbstunterweisung in Themen wie Programmieren und anderem geeignet ist.

Dies ist nur ein kurzer Überblick über die zur Verfügung stehenden Zusatzgeräte.

HALTEN SIE KONTAKT ZU IHREM COMMODORE-HÄNDLER, DER SIE ÜBER DIE NEUESTEN ENTWICKLUNGEN AUF DEM LAUFENDEN HÄLT.

Anhang B: Arbeiten mit Bandkassetten

Ein Kassetten-Rekorder ist sehr häufig das erste Peripherie-Gerät, welches von dem Benutzer an einen VC-20 angeschlossen wird. Dieses Gerät ermöglicht es, Programme für die spätere Verwendung auf Band zu speichern und Daten aufzubewahren, um diese zu einem späteren Zeitpunkt wieder lesen zu können. Dieser Anhang ist für diejenigen VC-20-Anlagen gedacht, die mit einem solchen Gerät ausgestattet sind.

Der erste im Zusammenhang mit dem Rekorder benutzte Befehl ist SAVE. Wird dieser Befehl VC-20 eingegeben, antwortet dieser mit folgender Meldung:

PRESS RECORD & PLAY ON TAPE
(Drücke die REC- und PLAY-Taste am Rekorder).

Jetzt muß die RECord-Taste des Kassettengeräts gehalten und die PLAY-Taste gedrückt werden. Beide Tasten sind jetzt eingerastet. (Läßt sich die RECord-Taste nicht herunterdrücken, dann sollte das Band daraufhin überprüft werden, ob die kleine Zarge am Bandgehäuse herausgebrochen wurde. Sollte dies der Fall sein, dann ist der VC-20 nicht in der Lage, auf dieses Band irgendwelche Informationen zu speichern. (Diese Zarge wird dann herausgebrochen, wenn es sich um ein Magnetband handelt, dessen Informationen und Daten besonders zu schützen sind.) Ist die PLAY-Taste gedrückt, dann erscheint die Meldung „OK" und danach das Wort SAVING (Speichern). Nach einer kurzen Zeit erscheint das Wort READY, was bedeutet, daß die Speicherungsaktion durchgeführt wurde. Es ist möglich, einem Programm einen Namen zuzuordnen, indem man nach dem Wort SAVE innerhalb von Anführungsstrichen (siehe Anhang C) einen Namen zufügt.

Nach Beendigung des Speichervorgangs ist es zu empfehlen, das gerade auf das Band geschriebene Programm zu überprüfen. Es könnte z. B. möglich sein, daß das Band selbst beschädigt war. Aus diesem Grund verfügt der VC-20 über den Befehl VERIFY. Dieser Befehl wird das auf dem Band befindende Programm mit dem im Hauptspeicher des VC-20 stehenden Programms vergleichen.

Spulen Sie also das Band zurück, und geben Sie VERIFY ein. Es erscheint folgende Meldung auf dem Bildschirm:

PRESS PLAY ON TAPE
(PLAY-TASTE BEIM BANDGERÄT DRÜCKEN).

Wenn zusätzlich zum Befehl VERIFY ein Programmname angegeben wird, sucht das System nach dem entsprechenden Programm. Wird dieser Programmname nicht spezifiziert, dann überprüft das System das erste Programm, welches sich auf dem Band befindet. Nach Betätigung der PLAY-Taste antwortet der VC-20 mit OK und dem Wort SEARCHING (Suchen). Danach werden alle Programme aufgelistet, deren Namen nicht mit dem hier eingegebenen übereinstimmen. Wurde das gesuchte Programm gefunden, dann erscheint die Nachricht FOUND (NAME) und VERIFYING. Jetzt wird das sich auf dem Band befindende Programm mit demjenigen verglichen, welches im Hauptspeicher geladen ist. Das gesamte Programm wird überprüft, auch

dann, wenn schon gleich am Anfang ein Fehler gefunden wird. Ist die Vergleichsoperation beendet, dann erscheint entweder die Nachricht OK und READY auf dem Bildschirm oder die Nachricht ?VERIFY ERROR. Bei einem solchen Fehler ist es zu empfehlen, ein anderes leeres Band zu verwenden, und die Operation SAVE und VERIFY zu wiederholen.

Soll ein Programm vom Magnetband in den Hauptspeicher des VC-20 geladen werden, dann ist der LOAD-Befehl einzusetzen. Der VC-20 meldet

PRESS PLAY ON TAPE
(PLAY-TASTE BEIM BANDGERÄT DRÜCKEN)

wonach das Programm geladen wird. Nach ordnungsgemäßer Beendigung des Ladevorgangs erscheint die Nachricht READY. Wurde ein Fehler im Ladevorgang entdeckt, dann erscheint die Meldung ?LOAD ERROR auf dem Bildschirm. Handelt es sich um ein mit vielen Mängeln behaftetes Band, dann könnten sich wundersame Dinge in dem VC-20 abspielen. So z. B. könnten recht fremdartige Zeichen auf dem Bildschirm erscheinen. In diesem Fall ist es zu empfehlen, den VC-20 ganz auszuschalten, und nach dem erneuten Einschalten (nach ein paar Sekunden) einen zweiten Versuch zu starten.

Damit sind wir an den etwas schwierigen Aspekten der Speicherung von Daten unter Verwendung von Bandkassetten angelangt (d. h., der Speicherung von Daten und der Ergebnisse aus Berechnungen), und selbstverständlich auch deren Wiederverwendung, also deren Einlesen in den Hauptspeicher.

Um Daten auf Band speichern zu können, muß zuerst eine Datei eröffnet werden. Dies wird durch den OPEN-Befehl erreicht, welchem 3 Zahlen und ein Name folgen. Dieser Befehl könnte wie folgt aussehen:

OPEN 1,1,1,"HALLO"

Die erste Zahl hinter dem OPEN-Befehl ist die Dateinummer, auf die sich das Programm bezieht. Die zweite Zahl bezieht sich auf das Peripherie-Gerät (bei einer Kassette eine „1"). Bei der dritten Zahl handelt es sich um einen Operations-Modus, wobei für Lesen eine „0", für Beschreiben eine „1" und für Beschreiben mit nachfolgendem Bandende-Kennzeichen eine „2" einzusetzen ist.

Der zwischen Anführungszeichen stehende Name (in diesem Fall HALLO) wird bei der Eröffnung zur Identifikation der Datei auf das Band geschrieben, wodurch es ermöglicht wird, viele Dateien mit unterschiedlichen Namen auf das Band zu bringen. Es ist hierbei zu bemerken, daß der VC-20 bei einer Schreiboperation nicht nach einem unbeschriebenen Teil des Bandes sucht, und daß der VC-20 bei einer Leseoperation das Band nach dem zur Verfügung gestellten Namen durchsucht. Die oben gezeigte OPEN-Anweisung wird eine Datei mit dem Namen HALLO auf dem Band erstellen, wobei diese Datei als Dateinummer 1 in anderen Programmen angesprochen werden muß.

Um Daten auf das Band zu bringen, ist die PRINT#-Anweisung zu verwenden. Dem Nummernzeichen (#: sprich FILE) folgt die entsprechende Dateinummer (im angezeigten Beispiel eine „1"), dann ein

Komma, und danach die Daten, die auf Band zu speichern sind. Diese Aneinanderreihung von Daten hat das gleiche Format wie bei Verwendung der PRINT-Anweisung im Zusammenhang mit der Bildschirmausgabe.

Wenn die Datei auf Band gespeichert wird, kann der Benutzer erkennen, daß das Band nicht fortwährend läuft, sondern nach Bedarf anhält und danach wieder weiterläuft. Dafür ist der Datenbuffer (Zwischenspeicher) des VC-20 verantwortlich, der die Daten auf ihrem Weg nach außen steuert. Das heißt, daß die Daten nicht direkt auf Band gespeichert, sondern zuerst in einen Spezialbereich des RAM-Speichers (Buffer) geschrieben werden. Dieser Buffer ist in der Lage, bis zu 192 Zeichen aufzunehmen. Wenn der Buffer voll ist, kann der VC-20 keine weiteren Zeichen darin speichern, und überträgt aus diesem Grund den gesamten Inhalt in einem Block auf das Band.

Nachdem alle Daten auf das Band gespeichert sind, muß die Datei unter Verwendung der CLOSE-Anweisung geschlossen werden. Der Befehl CLOSE muß unbedingt von der entsprechenden Dateinummer gefolgt sein. Ist dies nicht der Fall, dann könnte es möglich sein, daß die sich zu diesem Zeitpunkt im Buffer befindenden Zeichen verloren gehen. Nach der Übergabe der letzten Zeichen an das Band wird ein Bandende-Kennzeichen geschrieben, wenn dies auch entsprechend im OPEN-Befehl angegeben wurde (dritte Ziffer gleich „2").

Um die Daten wieder in den Hauptspeicher lesen zu können, muß zuerst der OPEN-Befehl für diese Datei eingegeben werden, wobei die dritte Ziffer nach dem OPEN eine „0" sein muß. Der VC-20 wird den Dateinamen suchen (wenn spezifiziert), wonach das Band anhält. Jetzt kann die INPUT#- oder GET#-Anweisung (sprich: INPUT-FILE/GET-FILE) eingesetzt werden, um die Daten in den Hauptspeicher zu bringen. Ist der Buffer zum Zeitpunkt der Ausführung der jeweiligen Anweisung leer, dann wird automatisch der zweite Datenteil bis zur Größe des Buffers (192 Zeichen) eingelesen. Die Daten werden ausnahmslos über den Buffer in den Hauptspeicher gelesen.

Der Unterschied zwischen INPUT# und GET# ist der, daß die INPUT#-Anweisung einen ganzen auf Band geschriebenen Datenblock übernimmt (egal, ob alpha-numerische Zeichen oder Zahlen), während GET# nur jeweils ein Zeichen übernimmt. Wenn sich der Benutzer die Mühe macht, mit der GET#-Anweisung zu experimentieren, wird er feststellen, daß sich hinter jedem letzten Zeichen eines Datenelements ein CHR$(13) befindet, welches dem Wagenrücklauf entspricht. Dieses Zeichen wird zur Aufteilung der sich auf Band befindenden Daten verwendet. Dadurch erkennt die INPUT#-Anweisung das Ende eines jeweiligen Datenblockes.

Die folgenden drei Beispielprogramme ermöglichen es dem Benutzer, das Lesen und Schreiben auf Band zu üben. Dabei erstellt das erste Programm die Datei. Nach der Ausführung des RUN-Befehls und dessen Beendigung muß das Band zurückgespult werden, wonach das zweite Programm ausgeführt werden kann. Dies demonstriert die Verwendung der INPUT#-Anweisung zur Wiedergewinnung (zum Lesen) der Daten. Das dritte Programm benutzt die GET#-Anweisung, womit die Daten Zeichen für Zeichen eingelesen werden. In diesem Programm wird sowohl das jeweilige Zeichen als auch dessen ASCII-Code auf dem Bildschirm dargestellt.

Programm 1: Programm schreiben auf Band

```
10 PRINT " [SHIFT] [CLR HOME] PROGRAMM SCHREIBEN AUF BAND"
20 OPEN 1,1,1,"DATEI"
30 PRINT "GIB JETZT DATEN"
35 PRINT "ZUR SPEICHERUNG EIN"
40 PRINT "ODER [CTRL] [RVS ON] STOP [CTRL] [RVS OFF]
50 PRINT
60 INPUT "DATEN";A$
70 PRINT#1,A$
80 IF A$ <> "STOP" THEN 50
90 PRINT
100 PRINT "DATEI WIRD GESCHLOSSEN"
110 CLOSE 1
```

Programm 2: Lesen von Band mit INPUT#-Befehl

```
10 PRINT " [SHIFT] [CLR HOME] PROGRAMM LESEN VOM BAND"
20 OPEN 1,1,0,"DATEI"
30 PRINT "DATEI ERÖFFNET"
40 PRINT
50 INPUT #1,A$
60 PRINT A$
70 IF A$ = "STOP" THEN CLOSE 1 : END
80 GOTO 40
```

Programm 3: Lesen von Band mit GET#-Befehl

Übernahme der Zeilen 10 bis 40 von Programm 2

```
50 GET#1,A$
60 IF A$ = "" THEN CLOSE 1 : END
70 PRINT A$,ASC(A$)
80 GOTO 50
```

Anhang C: VC-20 BASIC

Dieses Handbuch gab eine Einführung in die Programmiersprache BASIC, gerade so viel wie es nötig war, ein Gefühl für die Programmierung der Computer-Anlage und einige Teile des Vokabulars zu bekommen. Dieser Anhang gibt eine vollständige Liste von Regeln (SYNTAX) der Programmiersprache BASIC für den VC-20, zusammen mit einer knappen Beschreibung jeder einzelnen Regel. Es wird empfohlen, mit diesen Befehlen zu experimentieren, aber erinnern Sie sich, daß keiner der Befehle (auch keine fehlerhaften Eingaben) den VC-20 beschädigen kann, und daß die beste Lernmethode immer noch darin besteht, sich mit den eigentlichen Befehlen direkt an der Anlage auseinander zu setzen.

Dieser Anhang ist in Abschnitten entsprechend der unterschiedlichen Operationsarten von BASIC aufgeteilt. Diese beinhalten:

1. **Daten und Operatoren** – beschreibt unterschiedliche Datenarten der reservierten Datenfeld-Namen und der arithmetischen und logischen Operatoren.
2. **Befehle** – beschreibt die Befehle, die innerhalb der Programme, der Edit-Vorgänge und der Speicherung Verwendung finden sowie deren Löschung.
3. **Anweisungen** – beschreibt die BASIC-Programmieranweisungen, die in den numerierten Zeilen eines Programmes benutzt werden.
4. **Funktionen:** – beschreibt die Funktionen im Zusammenhang mit alpha-numerischen Zeichenketten (Strings), numerischen Werten und Ausgabeoperationen.

In jedem Abschnitt sind die Befehle aus Übersichtsgründen in alphabetischer Reihenfolge aufgelistet. Eine detailliertere Erklärung der BASIC-Befehle im Zusammenhang mit dem VC-20 befindet sich im VC-20 Programmierer-Handbuch.

1. Daten und Operatoren

a. Daten

Der VC-20 verwendet drei Arten von Daten innerhalb der BASIC-Programmiersprache. Diese sind wie folgt: *numerische Daten, ganzzahlige Daten und alpha-numerische Daten (Strings).*

Normale numerische Daten, auch als Gleitkomma-Werte bekannt, können einen Wert von 10^{-38} bis 10^{+38} haben, mit bis zu 9stelliger Genauigkeit. Wenn eine Zahl größer als 9 Ziffern ist, wie z. B. 10^{10} oder 10^{-10}, wird die Anlage diesen Wert in technisch/-wissenschaftliche Notation ausgeben, wobei die Zahl auf eine Stelle vor dem Komma und 8 Stellen nach dem Komma konvertiert ist, und der Buchstabe E mit der 10er Potenz folgt. So z. B. wird die Zahl 12345678901 als 1.23456789E + 11 dargestellt.

Ganzzahlige Werte werden verwendet, wenn die Zahl zwischen +32767 und −32768 liegt und keine Dezimalstellen vorhanden sind. Die ganzzahligen Werte benötigen weniger Hauptspeicher-Platz als Gleitkomma-Werte. Der Unterschied ist jedoch nicht sehr schwerwiegend, es sei denn, daß die Anzahl der ganzzahligen Werte sehr hoch ist, wie z. B. bei der Handhabung von Tabellen (siehe unten). Ein ganzzahliger Wert wäre z. B. eine Zahl wie 5, 10 oder −100.

Die *alpha-numerischen Zeichenketten* (Strings) sind Daten, die Zahlen, Buchstaben und alle Sonderzeichen des VC-20 enthalten können. Ein Beispiel einer alpha-numerischen Zeichenkette ist „VC-20".

Datennamen können aus einem einzelnen Buchstaben, einem Buchstaben gefolgt von einer Ziffer oder zwei Buchstaben bestehen.

Ein Datenname für einen ganzzahligen Wert wird unter Verwendung des *Prozentzeichens* (%) nach dem Datennamen definiert. Bei Datennamen für alpha-numerische Zeichenketten ist ein *Dollarzeichen* ($) hinter den Namen zu setzen.

BEISPIELE – Datenname für:
– numerische Werte: A, A5, BZ
– ganzzahlige Werte: A%, A5%, BZ%
– alpha-numerische Zeichenketten: A$, A5$, BZ$

Felder (ARRAYS) sind Aufstellungen von Daten mit gleichem Namen, wobei eine zusätzliche Zahl für das Einzelelement verwendet wird. Felder werden mit der DIM-Anweisung definiert und können Gleitkomma-Werte, ganzzahlige Werte oder alpha-numerische Daten enthalten. Dem Datenname eines Feldes folgt eine Zahl, die in runden Klammern einzuschließen ist, und das Einzelelement innerhalb des Feldes anspricht.

BEISPIELE: A(7),BZ%(11),A$(87).

In vielen Fällen werden 2-dimensionale Felder verwendet. Diese bestehen aus Reihen (horizontal) und Spalten (vertikal), wobei sich die erste Zahl innerhalb der Klammern auf die Reihe und die zweite Zahl auf die Spalte bezieht.

BEISPIELE: A(7,2)BZ%(2,3),Z%(3,2)

In dem VC-20 werden drei Datennamen benutzt, die ausschließlich für einen bestimmten Zweck reserviert sind und nicht für die Verwendung als normale Datennamen geeignet oder gedacht sind. Es handelt sich hierbei um die Namen ST, TI und TI$. ST ist eine Status-Variable, die sich auf Eingabe/Ausgabe-Operationen bezieht. In ST ist ein Wert enthalten, der geändert wird, wenn Probleme im Zusammenhang mit dem Laden von Programmen oder Daten von Band oder Diskette auftauchen. Eine detailliertere Beschreibung von ST befindet sich im BASIC-Programmiererhandbuch des VC-20.

TI und TI$ sind Werte, die sich auf die in dem VC-20 eingebaute Echtzeit-Uhr (real-time clock) beziehen. Der Wert TI wird jede 1/60stel Sekunde auf den neuesten Stand gebracht. Wenn der VC-20 eingeschaltet wird, fängt er mit 0 an und läuft bis zum Ausschalten kontinuierlich in 1/60stel Sekunden-Sprüngen.

In TI$ ist die Uhrzeit enthalten, die vom System automatisch verwaltet wird. Dabei stellen die ersten zwei Ziffern die Stunde, die nächsten 2 Ziffern die Minuten und die letzten 2 Ziffern die Sekunden dar. Diese Variable kann mit einem beliebigen Wert belegt werden, solange dieser numerisch ganzzahlig ist und genau 6 Stellen hat. Von diesem Zeitpunkt an wird er automatisch auf den neuesten Stand gebracht.

BEISPIEL: TI$ = "101530", setzt die Uhrzeit auf
10:15 und 30 Sekunden.

Beim Ausschalten des VC-20 wird die Uhrzeit gelöscht und fängt wieder mit 0 an, wenn der VC-20 erneut eingeschaltet wird.

b. Operatoren

Die arithmetischen Operatoren bestehen aus den folgenden Zeichen:

+	Addition
–	Subtraktion
*	Multiplikation
/	Division
↑	Potenzzeichen (Exponential-Rechnung)

Enthält eine Zeile mehr als einen Operator, dann werden die Rechenoperationen in einer bestimmten Reihenfolge ausgeführt. Dafür legt das System die folgenden Prioritäten fest: zuerst die Exponential-Rechnung, danach Multiplikation und Division und zuletzt die Addition und Subtraktion. Soll die Priorität (Rangordnung) geändert werden, dann muß ein Teil der Berechnung durch runde Klammern isoliert wird. Die sich in runden Klammern befindenden Rechenoperationen werden zuerst ausgeführt. Dabei ist sicherzustellen, daß bei Eingabe der Rechenformeln die gleiche Anzahl von linksbündigen und rechtsbündigen Klammern eingegeben werden. Ist dies nicht der Fall, dann wird zum Zeitpunkt der Ausführung die Meldung SYNTAX ERROR (syntaktischer Fehler) erscheinen.

Es sind auch Vergleichsoperatoren vorhanden:

=	gleich
<	kleiner als
>	größer als
< = oder = <	kleiner als oder gleich
> = oder = >	größer als oder gleich
< > oder > <	nicht gleich (ungleich)

Es stehen drei logische Operatoren zur Verfügung:

AND OR NOT

Am häufigsten werden diese Operatoren in verschachtelten Formeln IF . . . THEN verwendet.

BEISPIELE: IF A = B AND C = D THEN 100
erwartet, daß A = B & C = D wahr ist.
IF A = B OR C = D THEN 100
erwartet, daß entweder A = B oder C = D wahr ist.

2. Befehle

CONT (continue = fortsetzen)

Dieser Befehl wird für die Fortsetzung der Ausführung eines Programmes eingesetzt, welches entweder durch die Betätigung der STOP-Taste, durch das Einfügen einer STOP-Anweisung oder einer END-Anweisung ein Programm. Die Programmausführung wird genau dort wieder einsetzen, wo die Ausführung unterbrochen wurde.

CONT wird nicht ausgeführt, wenn Programmzeilen geändert oder hinzugefügt wurden (oder auch dann, wenn der Cursor lediglich zu einer Programmzeile gebracht und die RETURN-Taste gedrückt wurde, ohne eine Änderung einzugeben), wenn das Programm aufgrund eines Fehlers angehalten wurde oder wenn vom Benutzer vor dem Wiederanlauf des Programms ein Fehler verursacht wurde. In diesem Fall erscheint die Meldung CAN'T CONTINUE ERROR (KANN NICHT FORTSETZEN – FEHLER).

LIST

Der LIST-Befehl ermöglicht es, sich die Zeilen eines BASIC-Programmes anzusehen, welches sich zum Zeitpunkt im Hauptspeicher des VC-20 befindet. Der LIST-Befehl erzeugt eine vollständige Auflistung des Programmes auf dem Bildschirm (welche durch die Betätigung der CTRL-Taste gebremst oder durch die Betätigung der RUN/STOP-Taste vollkommen angehalten werden kann). Wird nach LIST eine Zeilennummer eingegeben, dann erscheint auf dem Bildschirm nur die entsprechende Programmzeile. Werden hinter LIST zwei Zahlen durch Bindestrich getrennt eingegeben, dann erscheinen auf dem Bildschirm alle zwischen diesen Zeilennummern liegenden Programmzeilen. Die Eingabe von LIST und einer Nummer, mit lediglich einem darauffolgenden Bindestrich, resultiert in der Bildschirmausgabe der Programmzeilen, angefangen von der definierten Zeile bis zum Ende des Programmes. Wird LIST, ein Bindestrich und danach eine Zahl eingegeben, dann erscheinen die Programmzeilen vom Anfang des Programms bis zur definierten Programmzeile. Unter Verwendung dieser verschiedenen Möglichkeiten ist es für den Benutzer sehr einfach, jeden einzelnen Teil eines Programmes zu untersuchen oder die entsprechende Programmzeile zum Zweck der Änderung auf den Bildschirm zu bringen.

BEISPIELE:	LIST	Zeigt das gesamte Programm
	LIST 10–	Zeigt Zeile 10 bis Programmende
	LIST 10	Zeigt nur Zeile 10
	LIST –10	Zeigt Zeilen, angefangen vom Start des Programms bis zur Zeile 10
	LIST 10–20	Zeigt die Zeilen 10 bis 20 (einschl.)

LOAD Abkürzung L SHIFT O

Dieser Befehl wird zum Laden eines Programmes benötigt, das sich auf Band oder Diskette befindet. Wird nur der LOAD-Befehl eingegeben und durch Drücken der RETURN-Taste bestätigt, sucht sich der VC-20 das erste auf dem Band gespeicherte Programm und bringt es in den Hauptspeicher. Es ist auch möglich, den LOAD-Befehl zusammen mit einem Programmnamen einzugeben, der sich innerhalb von Anführungszeichen befinden muß. Der Name kann noch von einem Komma (außerhalb Anführungszeichen) und einer Zahl oder von einem sich auf ein numerisches Feld beziehenden Namen gefolgt sein, die oder der in Verbindung mit einer Gerätenummer steht, von der das Programm geladen werden soll. Wurde keine Gerätenummer zur Verfügung gestellt, dann nimmt der VC-20 #1 an, und adressiert somit einen Bandkassetten-Rekorder.

Die andere sehr häufig in Verbindung mit dem LOAD-Befehl benutzte Gerätenummer ist #8, welche auf ein Disketten-Laufwerk zeigt.

BEISPIELE:

LOAD	Liest das sich auf Band befindende nächste Programm.
LOAD "HALLO"	Sucht auf Band nach dem Programm mit dem Namen HALLO und bringt dieses in den Hauptspeicher, nachdem es gefunden wurde.

LOAD A$ — Sucht nach einem Programm, dessen Name sich im Feld A$ befindet.

LOAD "HALLO",8 — Sucht nach einem Programm mit dem Namen HALLO, welches auf einer Diskette gespeichert ist.

LOAD "*",8 — Sucht nach dem ersten Programm auf der Diskette.

Der LOAD-Befehl kann innerhalb eines BASIC-Programmes angegeben werden, um das nächste sich auf Band befindende Programm zu laden und mit RUN auszuführen.

NEW

Dieser Befehl löscht das sich derzeitig im Hauptspeicher befindende Programm, und darüber hinaus alle Daten, die evtl. dort noch gespeichert sind. Das Programm ist gelöscht und kann nicht wieder aufgerufen werden, es sei denn, daß es vorher auf einem geeigneten Datenträger gespeichert wurde.
VORSICHT bei dem Einsatz dieses Befehles!

Der NEW-Befehl kann auch als Anweisung innerhalb eines BASIC-Programmes eingebettet werden. Kommt die VC-20-Anlage zu dieser Zeile, dann wird das Programm gelöscht und das System kommt zum Stillstand. Der Einsatz dieses Befehls ist dann sehr nützlich, wenn der Speicher nach der Ausführung des Programms „sauber" sein soll.

RUN

Nach der Eingabe eines Programms in den Hauptspeicher des Systems oder nach dem LOAD-Vorgang, kann der RUN-Befehl abgesetzt werden, wonach die Ausführung beginnt. Wird nur der RUN-Befehl eingegeben, dann fängt das System bei der niedrigsten Zeilennummer des Programms an. Folgt diesem Befehl eine Nummer, dann wird dem System die entsprechende Zeilennummer zum Start des Programms angezeigt.

BEISPIELE:

RUN — Fängt mit der Ausführung des Programms bei der niedrigsten Zeilennummer an.

RUN 100 — Fängt mit der Ausführung bei Zeilennummer 100 an.

RUN X — SYNTAX ERROR! (Es darf entweder RUN oder RUN mit einer Zeilennummer eingegeben werden – ein Buchstabe *an Stelle einer Zahl* führt zu einem Fehler).

SAVE

Dieser Befehl bringt ein sich derzeitig im Hauptspeicher befindendes Programm auf Band oder Diskette. Wird nur der SAVE-Befehl eingegeben und RETURN gedrückt, dann wird der VC-20 dieses Programm auf Band speichern. Er kann nicht überprüfen, ob sich an dieser Stelle schon ein Programm befindet. Deswegen ist dieser Befehl mit Umsicht einzusetzen. Folgt dem SAVE-Befehl ein in Anführungszeichen eingeschlossener Programmname, oder ein auf ein alpha-numerisches Feld weisender Name, dann wird diesem Programm der jeweilige Name zugeordnet, was es später ermöglicht, das jeweilige Programm leichter wieder aufzufinden und in den Hauptspeicher zu laden. Hinter dem Namen (nach dem letzten Anführungszeichen) kann eine Zahl oder ein Datenfeld-Name stehen. Dabei wird ein bestimmtes Peripherie-Gerät angesprochen, in welchem sich der Datenträger befindet, auf den das

Programm zu speichern ist. #1 bedeutet Kassetten-Rekorder und #8 Disketten-Laufwerk. Danach kann noch ein Komma und eine zweite Zahl folgen, die entweder 0 oder 1 ist. Bei 1 wird ein Bandende-Zeichen gesetzt. Soll ein Programm geladen werden und der VC-20 liest dieses Kennzeichen, dann erscheint die Nachricht FILE NOT FOUND ERROR (DATEI NICHT GEFUNDEN – FEHLER).

BEISPIEL:

SAVE	Speicherung des Programmes auf Band ohne Name
SAVE "HALLO"	Speicherung des Programmes auf Band mit Name HALLO
SAVE A$	Speicherung des Programmes mit Name in Datenfeld A$
SAVE "HALLO",8	Speicherung des Programmes auf Diskette mit Name HALLO
SAVE "HALLO",1,1	Speicherung des Programmes auf Band mit Name HALLO und Setzen eines Bandende-Kennzeichens nach Beendigung des Speichervorgangs.

VERIFY

Dieser Befehl vergleicht das sich im Hauptspeicher befindende Programm mit dem gerade abgelegten Programm. Er stellt sicher, daß der SAVE-Befehl ordnungsgemäß ausgeführt wurde und das Programm so auf dem Band oder der Diskette gespeichert ist. Dieser Befehl ist auch für die Positionierung eines Bandes sehr gut geeignet, so daß der VC-20 ein weiteres Programm *nach* dem zuletzt gespeicherten Programm auf Band aufzeichnen kann. Nach der Eingabe eines völlig neuen Programmes und dem Absetzen des VERIFY-Befehls sucht das System durch das Band und benachrichtigt den Benutzer davon, daß dieses Programm nicht zu finden ist (logisch). Jetzt befindet sich das Band an einer Position, von der ab das nächste Programm gespeichert werden kann, ohne befürchten zu müssen, daß ein noch gültiges Programm überschrieben wird.

Der VERIFY-Befehl ohne weitere Angabe vergleicht das erste auf Band gespeicherte Programm gegen das sich im Hauptspeicher befindende Programm. Wird VERIFY, gefolgt von einem sich in Anführungszeichen befindenden Programmnamen oder einem Datenfeld-Namen abgesetzt, dann wird auf dem Band nach dem definierten Programm gesucht. Danach wird die Vergleichsoperation ausgeführt. Wird zusätzlich ein Komma und eine Nummer (außerhalb der Anführungszeichen) angegeben, dann bezieht sich das System nicht nur auf den eingegebenen Programmnamen, sondern auch zusätzlich auf eine bestimmte Gerätenummer (1 für Band, 8 für Diskette).

BEISPIEL:

VERIFY	Vergleicht das nächste Programm auf Band
VERIFY "HALLO"	Sucht nach Programm HALLO und vergleicht mit Hauptspeicher-Inhalt
VERIFY "HALLO",8	Sucht nach Programm HALLO auf Diskette und vergleicht mit Hauptspeicher-Inhalt

3. Anweisungen

CLOSE

Dieser Befehl vervollständigt und schließt eine Datei ab, die durch den OPEN-Befehl eröffnet wurde. Die der CLOSE-Anweisung folgende Zahl bezieht sich auf die abzuschließende Datei.

BEISPIEL:

CLOSE 2	Nur Datei #2 ist abzuschließen.

CLR

Dieser Befehl löscht alle sich im Hauptspeicher befindenden Daten, läßt jedoch das Programm bestehen. Dieser Befehl wird automatisch durchgeführt, wenn RUN eingegeben wird.

CMD

Dieser Befehl überträgt die Datenausgabe, die normalerweise auf Bildschirm erfolgen würde (d.h., PRINT-Anweisungen, LISTs, jedoch keine POKEs auf ein anderes Gerät. Dabei könnte es sich um Drucker oder Dateien (Band oder Diskette) handeln. Diese Datei bzw. auch das Gerät muß zuerst unter Verwendung des OPEN-Befehls eröffnet werden. Dem CMD-Befehl muß entweder eine Zahl oder sich auf ein numerisches Feld beziehender Name folgen, der auf die entsprechende Datei hinweist.

BEISPIEL:

OPEN 1,4	Eröffnung des Gerätes #4 (Drucker)
CMD 1	Alle normalen Ausgaben erfolgen auf dem Drucker
LIST	Die Auflistung erfolgt nun auf dem Drucker und nicht auf dem Bildschirm – sogar das Wort LIST, welches eben eingegeben wurde.

Um die Ausgaben wieder auf den Bildschirm zu bringen, ist lediglich die Datei mit dem CLOSE-Befehl abzuschließen.

DATA

Dieser Anweisung folgt eine Ansammlung von Datenelementen, die bei der READ-Anweisung Verwendung finden. Die sich hier befindenden Einzelelemente können aus Zahlen oder Wörtern bestehen und werden durch Kommas getrennt. Es ist nicht notwendig, die Wörter in Anführungszeichen zu setzen, es sei denn, daß diese einen oder mehrere der folgenden Zeichen beinhalten: Leerzeichen, Doppelpunkt oder Komma. Wenn kein Wert zwischen zwei Kommas definiert wurde, dann wird dies bei numerischer Interpretation als 0 gewertet, oder bei einer alpha-numerischen Zeichenkette (String) als Leerzeichen.

BEISPIEL EINER DATA-ANWEISUNG:

DATA 100,200,FRIEDA,"HALLO, MUTTCHEN",3.14,abc123

Da es ein Programm nie notwendig hat, eine DATA-Anweisung für das Einlesen von Informationen auszuführen, ist es zu empfehlen, diese Anweisungen so weit wie möglich an das Ende des Programms zu setzen. Dadurch wird die Programmausführung beschleunigt.

DEF FN (Funktionsdefinition = Define Function)

Dieser Befehl ermöglicht es, eine sehr komplexe Berechnungsformel als Funktion zu definieren und diese mit einem Kurznamen auszustatten. Wird eine sehr lange Formel an verschiedenen Plätzen innerhalb des Programmes benötigt und definiert, kann dies viel Zeit bei der Programmentwicklung sparen.

Der hier einzusetzende Name besteht aus den Buchstaben FN, worauf ein nicht reservierter Feldname (eins bis zwei Zeichen lang) zu folgen hat. Zuerst ist die Funktion DEF zu spezifizieren, wonach der ausgewählte Name für diese Funktion erfolgt. Danach kommt ein, sich auf ein numerisches Feld beziehender Name (in diesem Fall X), der in Klammern zu setzen ist. Als letztes ist ein Gleichheitszeichen (=) zu setzen, worauf die entsprechende Formel eingegeben werden kann. Es ist jetzt möglich, diese einmal definierte Formel immer wieder aufzurufen, wobei die in X stehende Zahl ersetzt wird. Der Aufruf dieser Formel ist in Zeile 20 als Beispiel angegeben:

BEISPIEL:

```
10 DEF FNA(X) = 12*(34.75-X/.3)
20 PRINT FNA(7)
```


DIM (Felddimensionierung – Dimension an array)

Wenn mehr als 11 Elemente eines Variablenfeldes zu verarbeiten sind, muß vor der Feldverarbeitung zuerst eine DIM-Anweisung für dieses Feld ausgeführt werden. Hinter der DIM-Anweisung folgt der Feldname, der ein beliebiger, nicht reservierter Datenname sein kann. Danach folgt die in runden Klammern eingeschlossene Anzahl (oder der sich auf ein numerisches Feld beziehende Datenname) der Elemente einer jeden Dimension. Eine aus mehreren Dimensionen bestehendes Feld wird Matrix genannt. Es ist möglich, eine Vielzahl von Dimensionen zu verarbeiten. Dabei muß jedoch daran erinnert werden, daß die so in den Hauptspeicher gestellten Datenfelder viel Platz einnehmen. Wenn nicht mit größter Sorgfalt gearbeitet wird, ist es sehr leicht, die Kapazität des Hauptspeichers zu überschreiten. Zur Errechnung der Anzahl der Datenfelder, die mit jedem DIM-Befehl erstellt werden, ist lediglich die gesamte Zahl der Elemente mit jeder einzelnen Dimension der Tabelle zu multiplizieren.

BEISPIEL:

```
10 DIM A$(40),B7(15),CC%(4,4,4)
```


Es ist möglich, innerhalb einer DIM-Anweisung mehr als ein Feld zu spezifizieren, indem die einzelnen Feldnamen durch Kommas getrennt werden. Dabei ist darauf zu achten, daß die DIM-Anweisung für jedes Feld nur einmal ausgeführt werden darf. Wird diese Regel nicht beachtet, dann erscheint eine Fehlermeldung (REDIM'D ARRAY) auf dem Bildschirm. Im allgemeinen ist zu empfehlen, die DIM-Anweisungen an den Anfang des Programmes zu bringen.

END

Wenn ein Programm zur END-Anweisung kommt, dann hört der VC-20 mit der Verarbeitung auf, so, als ob keine Zeilen mehr zur Verfügung stünden. Das Programm kann unter Verwendung des CONT-Befehls wieder neu gestartet werden.

FOR . . . TO . . . STEP

Dieser Befehl ergibt in Verbindung mit dem NEXT-Befehl einen Programmabschnitt, dessen Befehle entsprechend der jeweiligen Definition häufig ausgeführt werden. So z. B. kann es sich lediglich darum handeln, daß der VC-20 damit beauftragt wird, in Schritten von eins bis zu einem bestimmten maximalen Schwellenwert zu zählen, nur um damit eine Zeitverzögerung einzuleiten. Andererseits kann es sich um eine Auszählung bzw. Berechnung handeln. Diese Anweisungen sind vielleicht die am häufigsten benutzten Befehle in der BASIC-Programmiersprache.

Das Format der Anweisung ist wie folgt:
FOR (Laufvariable) = (Startwert) TO (Endwert).

Die hier angegebene Laufvariable wird während des Programmlaufs verändert (Addition oder Subtraktion). Bei dem Start- und Endwert handelt es sich einerseits um den Wert von dem ab gezählt wird und andererseits um den Maximalwert bzw. Minimalwert (bis zu dem gezählt wird).

Die Logik der FOR-Anweisung ist wie folgt: zuerst wird die definierte Variable auf den Anfangswert gesetzt. Der Endwert wird von dem VC-20 für den späteren Vergleich aufbewahrt. Erreicht der VC-20 bei seiner Programmausführung eine Zeile mit dem Befehl NEXT, dann wird der Wert 1 dem Wert des Datenfeldes hinzuaddiert und das System überprüft, ob der nun erreichte Wert den definierten Endwert überschritten hat.

Ist dies nicht der Fall, dann wird die unmittelbar der FOR-Anweisung folgende Zeile ausgeführt. Hat der sich in der Variable befindende Wert den maximalen Schwellenwert überschritten, dann wird der sich unmittelbar hinter dem NEXT-Befehl befindende Befehl ausgeführt.

BEISPIEL:

```
10 FOR L = 1 TO 10
20 PRINT L
30 NEXT L
40 PRINT "ICH BIN FERTIG! L = "L
```

Dieses Programm wird die Zahlen von 1-10 auf den Bildschirm bringen, wonach die Nachricht ICH BIN FERTIG! L = 11 erscheint. Ist die Arbeitsweise verständlich? Sollte dies nicht der Fall sein, dann ist es zu empfehlen, den sich direkt vor dem Beispiel befindenden Absatz nochmals gründlich durchzulesen und das Programmbeispiel Schritt für Schritt in seiner Ausführung nachzuvollziehen.

Es ist möglich, direkt hinter dem Wort STEP eine zweite Zahl oder einen sich auf ein numerisches Feld beziehenden Datennamen zu stellen. In diesem Fall wird der hier eingetragene Wert oder der sich im Datenfeld befindende Wert anstatt der „1" addiert. Diese Einrichtung ermöglicht es rückwärts zu zählen, oder Zahlen mit Dezimalstellen zu verwenden.

Es ist auch möglich, ineinander verschachtelte Programmschleifen zu spezifizieren. Es muß hierbei *besonders darauf geachtet* werden, daß

die zuletzt angesprochene Programmschleife auch wieder als erstes verlassen wird.

BEISPIEL VERSCHACHTELTER PROGRAMMSCHLEIFEN:

```
10 FOR L = 1 TO 100
20 FOR A = 5 TO 11 STEP 2
30 NEXT A
40 NEXT L
```

Diese Programmschleife ist in die andere Programmschleife (Zeilen 10-40) eingebettet.

Falsch:

```
10 FOR L = 1 TO 100
20 FOR A = 5 TO 11 STEP 2
30 NEXT L
40 NEXT A
```

GET

Mit der GET-Anweisung ist es möglich, Daten von der Tastatur Zeichen für Zeichen zu übernehmen. Bei der Ausführung des GET-Befehls wird das gerade eingegebene Zeichen in den Speicher übernommen. Wurde kein Zeichen eingegeben, dann wird vom System eine Null (oder Leerstelle) empfangen und das Programm fährt mit der Verarbeitung fort. Es ist nicht notwendig, die RETURN-Taste zu betätigen. In der Tat ist es sogar so, daß auch das Drücken der RETURN-Taste als Signal in dem VC-20 empfangen wird.

Dem Wort GET folgt ein Datenfeld-Name, wobei das Feld normalerweise alpha-numerisch definiert wurde. Bei rein numerischer Spezifizierung kommt das Programm bei Betätigung einer nicht-numerischen Taste in Schwierigkeiten und wird deswegen zu einem fehlerhaften Abbruch führen. Die GET-Anweisung kann in einer Programmschleife eingebettet werden, mit der Absicht, daß sich das Programm so lange in dieser Schleife bewegt, bis eine richtige Eingabe erfolgt ist. Das Programm fährt dann mit seiner Verarbeitung fort.

GET #

Dieser Befehl wird in Verbindung mit einer vorher eröffneten Datei (OPEN) benutzt, wobei die Zeichen separat in den Hauptspeicher des Systems gelangen.

BEISPIEL: GET#1,A#

GOSUB . . . RETURN

Dieser Befehl ist in seiner Auswirkung ähnlich der GOTO-Anweisung, mit der Ausnahme, daß der VC-20 daran erinnert, von welcher Stelle aus verzweigt wurde. Wird eine Zeile verarbeitet, welche die RETURN-Anweisung enthält, dann springt das Programm zurück zu der Stelle, die unmittelbar der GOSUB-Anweisung folgt. Dieser Befehl läßt sich dann besonders gut einsetzen, wenn eine Routine mehrere Male auszuführen ist. Anstatt diese Routine immer wieder neu eingeben zu müssen, ist die

Eingabe nur ein einziges Mal erforderlich, wonach unter Verwendung der GOSUB-Anweisung auf die entsprechende Routine mehrmals verzweigt werden kann. 20 GOSUB 800 bedeutet, daß die Unterroutine, welche bei Zeile 800 beginnt, auszuführen ist.

GOTO oder GO TO

Wird eine Anweisung mit dem GOTO-Befehl erreicht, dann wird die Programmausführung bei der Zeile fortsetzen, die dem Befehl GOTO unmittelbar folgt.

IF ... THEN

Die Anweisung IF ... THEN analysiert eine bestimmte Situation, wonach zwei mögliche Richtungen, abhängig von dem Resultat der Analyse, eingeschlagen werden können. Ist die Bedingung wahr, wird der Befehl angesprochen, der THEN direkt folgt. Dies kann selbstverständlich eine Zeilennummer sein, was dazu führt, daß der VC-20 zu den in dieser Zeile definierten Befehl verzweigt. Es kann auch eine ganze Reihe von BASIC-Anweisungen folgen. Ist die Bedingung falsch, dann wird die nächste Zeile (nicht der nächste Befehl auf der gleichen Zeile) stattdessen ausgeführt.

Der zu vergleichende Teil kann aus einem Datenwert oder einer Formel bestehen. In den meisten Fällen wird ein *logischer Vergleich angestellt* (=, <, >, AND, OR, NOT). Siehe Abschnitt der Bedingungsoperatoren für die Erklärung der Vorgehensweise.

INPUT

Die INPUT-Anweisung ermöglicht die Eingabe variabler Daten über die Tastatur. Das Programm wird nach der Anzeige eines Fragezeichens (?) auf dem Bildschirm mit seiner Ausführung anhalten und auf die Antwort des Benutzers warten, die nach Betätigung der RETURN-Taste vom System empfangen wird.

INPUT wird mindestens von einem Variablen-Namen gefolgt. Es können auch mehrere sein (durch Kommas getrennt). Es ist auch gestattet, eine sich innerhalb von Anführungszeichen befindende Nachricht zwischen INPUT und die Datenfeld-Namen zu setzen. Wird diese Nachricht kodiert, dann muß dem letzten Anführungszeichen der Nachricht ein Strichpunkt (;) folgen.

BEISPIEL:

```
10 INPUT"BITTE IHR ALTER EINGEBEN";A
20 INPUT"...UND IHREN NAMEN";A$
30 PRINT A$" IST "A" JAHRE ALT"
```

INPUT#

Diese Anweisung arbeitet ähnlich der INPUT-Anweisung (oben), mit der Ausnahme, daß sie sich auf eine vorher eröffnete Datei bezieht.

LET

LET wird sehr selten in einem Programm verwendet, da dessen Einsatz wahlfrei ist. Die Anweisung selbst jedoch ist in den meisten Fällen das Herz eines jeden BASIC-Programmes. Der Variablen-Name, in dem das Ergebnis der Berechnung zu speichern ist, befindet sich auf der linken

Seite des Gleichheitszeichens (=) und die Zahl oder Formel auf dessen rechten Seite.

BEISPIEL:

```
10 LET A = 5
20 B = 6
30 C = A*B+3
40 D$ = "HALLO"
```

NEXT

Die NEXT-Anweisung wird stets in Verbindung mit der FOR-Anweisung eingesetzt. Wenn das Programm einen NEXT-Befehl erreicht, verzweigt es zurück zur FOR-Anweisung und überprüft die Programmschleifen-Aktivität (siehe FOR-Anweisung für weitere Informationen). Ist die Programmschleife beendet, dann wird die Programmausführung mit dem Befehl fortgesetzt, der sich direkt hinter der NEXT-Anweisung befindet. Das Wort NEXT wird von einem Variablen-Namen oder von einer Vielzahl verschiedener Variablen-Namen gefolgt, die in diesem Fall durch Kommas getrennt sein müssen. Sind keine Variablen-Namen vorhanden, dann ist die zuletzt angefangene Programmschleife beendet. Bei Angabe von mehreren Variablen-Namen werden diese der Reihe nach von links nach rechts abgehandelt.

BEISPIEL:

```
10 FOR L = 1 TO 10:NEXT
20 FOR L = 1 TO 10:NEXT L
30 FOR L = 1 TO 10:FOR M = 1 TO 10:NEXT M,L
```

ON

Mit diesem Befehl ist es möglich, die GOTO- und GOSUB-Anweisungen in eine Art Sonderversion der IF-Anweisung zu verwandeln. Dem Wort ON folgt ein Variablen-Name, der sich auf einen Wert bezieht. GOTO oder GOSUB wird von einer Aneinanderreihung von Zeilennummern gefolgt, die durch Kommas getrennt sind. Ist das Ergebnis der Berechnung gleich 1, dann wird die erste definierte Zeilennummer angesprungen. Bei einer 2 als Ergebnis wird auf die zweite definierte Zeilennummer verzweigt, usw. Ist das Ergebnis 0, negativ oder größer als die Anzahl der aufgeführten Zeilennummern, dann wird der Befehl als nächstes ausgeführt, der unmittelbar hinter der ON-Anweisung folgt.

BEISPIEL:

```
10 INPUT X
20 ON X GOTO 10,50,50.50
30 PRINT "NOPE!"
40 GOTO 10
50 PRINT"YUP!"
60 ON X GOTO 10,30,30
```

OPEN

Die OPEN-Anweisung ermöglicht es dem VC-20 auf Peripherie-Gerät wie z. B. Kassetten-Rekorder, Disketten-Laufwerke, Drucker oder sogar auf den Bildschirm des VC-20 zuzugreifen. Dem Wort OPEN folgt eine Zahl – die Kennummer des Files. Diese Zahl kann zwischen 1 und 255 liegen. Normalerweise befindet sich eine zweite Zahl hinter der ersten, wobei diese durch Komma getrennt sind. Bei der zweiten Zahl handelt es sich um die Gerätenummer, wobei 0 für den Bildschirm des VC-20,

1 für den Kassetten-Rekorder, 4 für den Drucker und 8 für Disketten-Laufwerk steht. Es ist zu empfehlen, die Bezugsnummer (erste Zahl) der Gerätenummer gleichzusetzen. Hinter der zweiten Zahl kann noch eine dritte Zahl stehen, die von dieser durch Komma getrennt ist. Dabei handelt es sich um eine Zusatzinformation. Wird eine Band-Kassette verwendet, dann bedeutet eine 0, daß das Band zu lesen ist, eine 1, daß das Band zu schreiben ist und eine 2, daß bei Programmende ein Bandende-Kennzeichen zu setzen ist. Beim Einsatz einer Diskette beziehen sich diese Zahlen auf Puffer oder Datenkanal. Adressiert dieser Befehl einen Drucker, dann stellt diese Zusatzinformation verschiedene Befehle dar. Wenn weitere Informationen gewünscht sind, dann ist das VC-20 Programmierer-Handbuch einzusehen. Als letztes kann noch eine sich in Anführungszeichen befindende alpha-numerische Zeichenkette eingetragen werden, die als Befehl bei Einsatz eines Disketten-Laufwerkes anzusehen ist oder einen Namen einer Datei im Zusammenhang mit der Speicherung von Daten auf Band zur Verfügung stellt.

BEISPIEL:

10 OPEN 1.0	Eröffnet den Bildschirm
20 OPEN 2,1,0,"D"	Eröffnet eine Band-Kassette zum Lesen einer Datei (Name = D).
30 OPEN 3,4	Einsatz eines Druckers.
40 OPEN 4,8,15	Eröffnet den Datenkanal, der die Verbindung zwischen Disketten-Laufwerk und System herstellt.

Siehe auch CLOSE, CMD, GET#, INPUT# und PRINT#, System-Datenfeld ST und Anhang B.

POKE

Dem POKE-Befehl folgen immer zwei Zahlen oder Formeln. Die erste Zahl bezieht sich auf eine bestimmte Stelle (ADRESSE) innerhalb des Hauptspeichers des VC-20. Man kann Hauptspeicher-Stellen von 0 bis über 65.000 ansprechen. Einige dieser Hauptspeicher-Stellen lassen sich leicht und problemlos innerhalb eines Programmes adressieren, wie z. B. im Kapitel über Ton und Farben gezeigt. Andere Hauptspeicher-Bereiche werden jedoch von dem VC-20 selbst benötigt, die von dort aus den Programmablauf überwachen. Experimente mit POKE-Anweisungen werden möglicherweise zu sehr eigenartigen Ergebnissen führen. Findet ein Ablauf innerhalb des Systems statt, der nicht von selbst anhält, dann muß entweder der VC-20 ausgeschaltet und danach wieder eingeschaltet werden, oder es ist zu versuchen, die RUN/STOP-Taste zu halten und die RESTORE-Taste zu betätigen.

Die zweite Zahl kann zwischen 0 und 255 liegen und überschreibt und ersetzt somit das Zeichen, welches sich in der jeweils angesprochenen Hauptspeicher-Position befindet.

BEISPIEL:

```
10 POKE 36879,8
20 POKE 9*16↑3+15,27
```

PRINT

Bei der PRINT-Anweisung handelt es sich möglicherweise um den ersten Befehl, den man im anfänglichen Lernprozess einsetzt. Es sind jedoch auch in diesem Zusammenhang einige Feinheiten zu beachten,

die auch gemeistert sein wollen. Dem Wort PRINT können folgende Dinge folgen:

- Wörter innerhalb von Anführungszeichen,
- Datenfeldnamen,
- Funktionen,
- Satzzeichen.

Die sich innerhalb von Anführungszeichen befindenden Wörter werden auch häufig „Zeichenketten" genannt. Bei der Angabe von Variablen-Namen außerhalb von Anführungszeichen wird der sich in dem entsprechenden Feld befindende Wert ausgegeben. Bei Funktionen wird auch der Wert der jeweiligen Berechnung ausgegeben. Die Satzzeichen dienen dazu, das Datenformat übersichtlich auf den Bildschirm zu bringen. Das Komma wird dazu verwendet, den Bildschirm in zwei Spalten aufzuteilen, während der Strichpunkt keine Leerstellen übrig läßt. Sowohl das Komma als auch der Strichpunkt können als Abschluß des PRINT-Befehls eingesetzt werden, woraus resultiert, daß sich der nächste PRINT-Befehl dieser PRINT-Anweisung direkt anschließt, so als ob es sich lediglich um eine Fortsetzung handelt.

BEISPIEL:

```
10 PRINT "HALLO"
20 PRINT "HALLO,"A$
30 PRINT A+B;
50 PRINT J;
60 PRINT A,B.C,D
```

Siehe auch Funktionen: POS(), SPC(), TAB()

PRINT #

Dieser Befehl unterscheidet sich etwas von der oben aufgeführten PRINT-Anweisung. Dem Befehl PRINT # folgt immer eine Zahl, die sich auf ein Peripherie-Gerät oder auf eine Datei bezieht, welche vorher durch OPEN eröffnet wurde. Die Zahl wird von einem Komma und danach von einer Aneinanderreihung von Daten gefolgt, die auszugeben sind. Das Komma und der Strichpunkt haben die gleiche Auswirkung hinsichtlich der Verwendung von Leerzeichen, wie dies auch schon bei PRINT beschrieben wurde. Bei einigen Peripherie-Geräten ist TAB und SPC nicht einsetzbar.

BEISPIEL:

```
100 PRINT#1,"HALLO FREUNDE!";A$,B$
```

READ

Dieser Befehl wird benutzt, um Informationen von DATA-Anweisungen in Datenfelder zu bringen, von wo aus sie weiter verwendet werden können. Es ist darauf zu achten, das Lesen von alpha-numerischen Zeichen dann zu vermeiden, wenn die READ-Anweisung eine Zahl erwartet. Nichtbeachtung dieser Regel führt zu dem Fehler TYPE MISMATCH ERROR (DATENFELD-FEHLER).

REM (Remark = Bemerkung)

REM-Eingaben sind keine Befehle als solches und dienen lediglich der Erklärung des Programms. Mit dieser Eingabe können Teile von Programmen beschrieben werden. REM hat keinerlei Auswirkung auf die Ausführung des Programms, wirkt sich jedoch auf die Programmlänge

aus. Dem Wort REM kann ein beliebiger Text folgen, wobei jedoch der Einsatz von graphischen Zeichen zu etwas befremdenden Ergebnissen führen könnte (siehe VC-20 Programmierer-Handbuch).

RESTORE

Innerhalb eines Programms verursacht dieser Befehl das Zurücksetzen des READ-„Zeigers" auf die erste DATA-Anweisung. Dadurch ist es möglich, Daten immer wieder von neuem zu lesen. RESTORE benötigt keine Zusatzinformationen.

RETURN

Diese Anweisung wird stets in Verbindung mit dem GOSUB-Befehl eingesetzt. Trifft das Programm auf RETURN, dann geht das System auf den Befehl zurück, welcher der GOSUB-Anweisung direkt folgt. Wurde vorher kein GOSUB-Befehl registriert, dann führt dies zur Meldung: RETURN WITHOUT GOSUB ERROR (RETURN OHNE GOSUB – FEHLER). RETURN benötigt keine Zusatzinformation.

STOP

Bei dieser Anweisung wird die Programmausführung angehalten. Danach folgt die Nachricht BREAK ERROR IN LINE xxxx (PROGRAMM-HALT IN ZEILE xxxx), wobei xxxx sich auf die Zeile bezieht, welche die STOP-Anweisung enthält. Durch Absetzen der CONT-Anweisung kann das Programm erneut gestartet werden. Der STOP-Befehl wird zur Programmfehler-Prüfung eingesetzt.

SYS

Auf das Wort SYS folgt entweder eine Dezimalzahl oder ein sich auf ein numerisches Feld beziehender Name, wobei der Wert zwischen 0 und 65535 liegen kann. Danach fängt das System mit der Ausführung eines in Maschinenkode geschriebenen Programms an, welches bei der hier definierten Hauptspeicher-Position beginnt. Dieser Befehl ist in seiner Auswirkung ähnlich der USR-Funktion, erlaubt es jedoch nicht, Parameter-Daten zu übertragen.

WAIT

Die WAIT-Anweisung wird dazu verwendet, den Programmablauf so lange anzuhalten, bis sich der Wert einer bestimmten Hauptspeicher-Position auf einen genannten Wert ändert. Dem Wort WAIT folgt eine Zahl, die sich auf die zu prüfende Hauptspeicher-Adresse bezieht. Danach folgt ein Komma und eine weitere Zahl. Es ist auch möglich, diesem Wert ein weiteres Komma und eine dritte Zahl folgen zu lassen. Die letzten zwei Zahlen können zwischen 0 und 255 liegen.

Der Inhalt der angesprochenen Hauptspeicher-Position wird zuerst mit der dritten Zahl (wenn anwesend) in einer exklusiven OR-Verknüpfung verarbeitet und danach mit einer logischen AND-Verknüpfung mit der zweiten Zahl. Wenn das Ergebnis gleich 0 ist, geht das Programm zur gleichen Hauptspeicher-Position zurück und prüft nochmals. Ist das Resultat nicht gleich 0, dann erfolgt die Ausführung des nächsten Befehls.

4. Funktionen

a. Numerisch

ABS(X) (absoluter Wert)

Diese Funktion gibt den absoluten Wert einer Zahl wieder, also ohne arithmetisches Vorzeichen (– oder +). Die Antwort ist immer positiv.

ATN(X) (arcustangens)

Gibt den Winkel in Bogenmaß an, dessen Tangens gleich X ist.

COS(X) (Cosinus)

Stellt den Wert des Cosinus von X zur Verfügung, wobei X den in Bogenmaß gemessenen Winkel darstellt.

EXP(X)

Potenziert die mathematische Konstante e (2,71828183) mit dem Wert X.

FNXX(X)

Stellt den Wert einer vom Benutzer definierten Funktion XX zur Verfügung, die in einer DEF FNXX-Anweisung definiert wurde.

INT(X) (Ganzzahl)

Stellt einen Wert aus X zur Verfügung, dessen Dezimalstellen abgetrennt wurden. Das Resultat wird also immer kleiner als oder gleich X sein. Daher wird aus jeder Zahl mit negativem Vorzeichen eine Ganzzahl, die *kleiner als* der eigentliche Wert ist.

Wird die INT-Funktion zur Auf- oder Abrundung benutzt, dann ist sie folgendermaßen einzusetzen: INT (X + .5).

BEISPIEL:

X = INT(X*100 + .5)/100 Rundet auf den nächsten Pfennig

INT(–5.5) = –6

LOG(X) (Logarithmus)

Mit dieser Funktion wird der natürliche Logarithmus von X zur Verfügung gestellt. Dabei handelt es sich um den Logarithmus zur Basis e (siehe EXP(X)). Zur Umrechnung auf Logarithmus zur Basis von 10 ist lediglich durch LOG(10) zu dividieren.

PEEK(X)

Diese Funktion wird dazu verwendet, den Inhalt einer bestimmten Hauptspeicher-Position (X) (0 bis 65535) zu überprüfen, wonach ein Wert von 0 bis 255 ausgegeben wird. Diese Funktion wird sehr häufig in Verbindung mit der POKE-Anweisung eingesetzt.

RND(X) (Zufallszahlen)

Diese Funktion stellt eine Zufallszahl zwischen 0 und 1 zur Verfügung. Sie wird hauptsächlich in Verbindung mit Glücksspielen (Würfelspiel) eingesetzt und ist auch im Zusammenhang mit einigen statistischen Applikationen behilflich. Die erste Zufallszahl sollte mittels der Formel RND(-TI) generiert werden, um zu gewährleisten, daß der Anfang immer unterschiedlich ist. Danach sollte der Wert von X gleich 1 oder gleich einer positiven Zahl sein.

Zur Simulation eines Würfelspiels ist die Formel INT(RND(1)*6+1) einzusetzen. Zuerst ist die willkürliche Zahl von 0 bis 1 mit 6 zu multiplizieren, woraus folgt, daß sich die Bandbreite auf zwischen 0 und 6 erweitert (tatsächlich größer als 0 und kleiner als 6). Dann wird der Wert 1 dazu addiert, woraus sich die Bandbreite 1 unter 7 ergibt. Die INT-Funktion trennt alle Dezimalstellen ab, was zu einem Ergebnis zwischen 1 und 6 führt.

Zur Simulation von 2 Würfeln sind die zwei durch obige Formel erhaltenen Zahlen zu addieren.

BEISPIEL:

100 X = INT (RND(1)*6)+INT(RND(1)*6+1	Simulation von 2 Würfeln
100 X = INT(RND(1)*1000)+1	Zahlen zwischen 1 und 1000
100 X = INT(RND(1)*150)+100	Zahlen zwischen 100 und 249

SGN(X) (Vorzeichen)

Diese Funktion stellt das jeweilige arithmetische Vorzeichen (positiv, negativ oder Null) von X zur Verfügung. Daraus resultiert +1 für positiv, 0 für Null und −1 für negativ.

SIN(X) (Sinus)

Hierbei handelt es sich um die trigonometrische Sinus-Funktion. Daraus ergibt sich Sinus von X, wobei X den in Bogenmaß gemessenen Winkel enthält.

SQR(X) (Wurzelziehen)

Diese Funktion stellt die Quadrat-Wurzel von X zur Verfügung, wobei X eine positive Zahl oder Null enthält. Ist X negativ, dann handelt es sich um einen Fehler (ILLEGAL QUANTITY ERROR).

TAN(X) (Tangens)

Diese Funktion stellt den Wert der Tangens von X zur Verfügung, wobei X den im Bogenmaß gemessenen Winkel enthält.

USR(X)

Unter Verwendung dieser Funktion springt das Programm zu einem in Maschinenkode geschriebenen Programm, dessen Anfang in den Hauptspeicher-Positionen 1 und 2 zu finden ist. Der Parameter X wird dem Maschinenkode-Programm übertragen, welches dem BASIC-Programm eine andere Zahl übergibt. Für weitere Informationen ist das VC-20 Programmierer-Handbuch einzusehen.

b. STRING-Funktionen

ASC(X$)

Diese Funktion stellt den ASCII-Kode des ersten Zeichens von X$ zur Verfügung.

CHR$(X)

Im Gegensatz zur Funktion ASC stellt diese Funktion ein alphanumerisches Zeichen zur Verfügung, dessen ASCII-Kode in X gefunden wird.

LEFT$(X$,X)

Diese Funktion stellt den linksbündigen Teil des Feldes X$ in der Länge von „X“ Zeichen zur Verfügung.

LEN(X$)

Diese Funktion ermittelt die Anzahl von Zeichen (einschließlich Leerzeichen und anderer Sonderzeichen) einer alpha-numerischen Zeichenkette (X$).

MID$(X$,S,X)

Diese Funktion stellt eine X-Anzahl von alpha-numerischen Zeichen, angefangen von der S-ten Stelle des Feldes X$, zur Verfügung.

RIGHT$(X$,X)

Diese Funktion stellt die X-Anzahl der rechtsbündigen Zeichen des Feldes X$ zur Verfügung.

STR$(X)

Diese Funktion wandelt den Wert X in eine Zeichenkette:

Beispiel:

X$ = STR$(X)

VAL(X$)

Diese Funktion wandelt die alpha-numerische Zeichenkette aus X$ in einen numerischen Wert um, im gewissen Sinne die Umkehrung der Funktion STR$. Dabei wird Zeichen für Zeichen von links nach rechts auf gültigen numerischen Inhalt überprüft, so lange, bis ein nicht numerisches Zeichen erreicht wird. Findet der VC-20 ungültige, nicht erlaubte Zeichen, dann wird nur ein Teil bis zu diesem Punkt konvertiert

Beispiel:

10 X = VAL("123.456)	X = 123.456
10 X = VAL("12A13B")	X = 12
10 X = VAL("RIU017*")	X = 0
10 X = VAL("−1.23.23.23")	X = −1.23

c. Andere Funktionen

FRE(X)

Diese Funktion zeigt die Größe des nicht belegten Hauptspeicher-Bereiches an, unabhängig von dem in der Variablen X gespeicherten Wert.

POS(X)

Diese Funktion zeigt die Spalte an, in der die nächste PRINT-Anweisung auf dem Bildschirm wirksam wird, unabhängig von dem in X genannten Wert.

SPC(X)

Diese Funktion wird in Verbindung mit der PRINT-Anweisung eingesetzt und verursacht den Vorschub der nächsten Ausgabezeile um X Spalten.

TAB(X)

Diese Funktion wird in Verbindung mit der PRINT-Anweisung eingesetzt und verursacht die Ausgabe des nächsten Zeichens auf Spalte X.

Anhang D: Abkürzungen der BASIC-Schlüsselwörter

Um Zeit zu gewinnen und der Einfachheit halber ist es mit der BASIC-Programmiersprache des VC-20 erlaubt, für fast alle Schlüsselwörter Abkürzungen zu verwenden. Die Abkürzung für den Befehl PRINT ist ein Fragezeichen (?). Die Abkürzung der anderen Schlüsselwörter besteht entweder aus dem ersten relevanten Buchstaben oder, wenn dies nicht eindeutig genug ist, aus den ersten zwei Buchstaben, wonach der jeweilige nächste Buchstabe im Umschalt-Modus (SHIFT-Taste) einzugeben ist. Erscheinen diese Abkürzungen innerhalb einer Programmzeile, dann wird der vollständige Befehl unter Verwendung des Befehls LIST erscheinen.

Kommando	Schlüssel	Bildschirm-Bild
AND	A SHIFT N	A
NOT	N SHIFT O	N
CLOSE	CL SHIFT O	CL
CLR	C SHIFT L	C
CMD	C SHIFT M	C
CONT	C SHIFT O	C
DATA	D SHIFT A	D ♠
DEF	D SHIFT E	D
DIM	D SHIFT I	D
END	E SHIFT N	E
FOR	F SHIFT O	F
GET	G SHIFT E	G
GOSUB	GO SHIFT S	GO ♥
GOTO	G SHIFT O	G
INPUT#	I SHIFT N	I
LET	L SHIFT E	L
LIST	L SHIFT I	L
LOAD	L SHIFT O	L
NEXT	N SHIFT E	N
OPEN	O SHIFT P	O
POKE	P SHIFT O	P
PRINT	?	?
RND	R SHIFT N	R
SGN	S SHIFT G	S
SIN	S SHIFT I	S
SPC (	S SHIFT P	S
SQR	S SHIFT Q	S ●
PRINT#	P SHIFT R	P
READ	R SHIFT E	R
RESTORE	RE SHIFT S	RE ♥
RETURN	RE SHIFT T	RE
RUN	R SHIFT U	R
SAVE	S SHIFT A	S ♠
STEP	ST SHIFT E	ST
STOP	S SHIFT T	S
SYS	S SHIFT Y	S
THEN	T SHIFT H	T
VERIFY	V SHIFT E	V
WAIT	W SHIFT A	W ♠
ABS	A SHIFT B	A
ASC	A SHIFT S	A ♥
ATN	A SHIFT T	A
CHR$	C SHIFT H	C
EXP	E SHIFT X	E ♣
FRE	F SHIFT R	F
LEFT$	LE SHIFT F	LE
MID$	M SHIFT I	M
PEEK	P SHIFT E	P
RIGHT$	R SHIFT I	R
STR$	ST SHIFT R	ST
TAB (	T SHIFT A	T ♠
USR	U SHIFT S	U ♥
VAL	V SHIFT A	V ♠

Anhang E: Farbkombinationen des Bildschirm-Hintergrunds und -Rahmens

Es ist möglich, die Farben des Bildschirm-Hintergrunds und -Rahmens entweder ausgehend von einem Programm oder außerhalb eines Programmes zu ändern. Dafür ist folgender Befehl einzugeben:

POKE 36879,X

wobei X eine Zahl ist, die aus der unten stehenden Tabelle ausgewählt werden kann. POKE 36879,27 bringt den Bildschirm zur Normalfarbe zurück, was bedeutet, daß der Bildschirm-Rahmen hellblau und der Hintergrund weiß wird.

Gibt man z. B. den Befehl POKE 36879,8 ein und drückt danach 

und [2 WHT] dann erscheinen weiße Buchstaben auf vollkommen schwarzem Hintergrund! Der Benutzer sollte jetzt einige andere Kombinationen austesten. Der POKE-Befehl ist die schnellste und einfachste Lösung, eine Änderung der Bildschirmfarben vorzunehmen.

	Bildschirmrahmen							
	BLK	WHT	RED	CYAN	PUR	GRN	BLU	YEL
Hintergrund	**SCHW**	**WEISS**	**ROT**	**TÜRKIS**	**VIOLETT**	**GRÜN**	**BLAU**	**GELB**
SCHWARZ	8	9	10	11	12	13	14	15
WEISS	24	25	26	27	28	29	30	31
ROT	40	41	42	43	44	45	46	47
TÜRKIS	56	57	58	59	60	61	62	63
VIOLETT	72	73	74	75	76	77	78	79
GRÜN	88	89	90	91	92	93	94	95
BLAU	104	105	106	107	108	109	110	111
GELB	120	121	122	123	124	125	126	127
ORANGE	136	137	138	139	140	141	142	143
HELLORAN.	152	153	154	155	156	157	158	159
ROSA	168	169	170	171	172	173	174	175
HELLTÜRKIS	184	185	186	187	188	189	190	191
HELLVIOL.	200	201	202	203	204	205	206	207
HELLGRÜN	216	217	218	219	220	221	222	223
HELLBLAU	232	233	234	235	236	237	238	239
HELLGELB	248	249	250	251	252	253	254	255

Anhang F: Tabelle der Musiknoten

NOTE	WERT	NOTE	WERT
C	135	G	215
C#	143	A^b	217
D	147	A	219
E^b	151	B	221
E	159	H	223
F	163	C	225
F#	167	C#	227
G	175	D	228
A^b	179	E^b	229
A	183	E	231
B	187	F	232
H	191	F#	233
C	195	G	235
C#	199	A^b	236
D	201	A	237
E^b	203	B	238
E	207	H	239
F	209	C	240
F#	212	C#	241

Stimmlagen-Befehle	X =	Funktion
POKE 36878, X	0 bis 15	setzt die Lautstärke
POKE 36874, X	128 bis 255	spielt Note
POKE 36875, X	128 bis 255	spielt Note
POKE 36876, X	128 bis 255	spielt Note
POKE 36877, X	128 bis 255	spielt Sondereffekte

Anhang G: 20 Sondereffekte des VC-20

Es folgen einige Beispiele, die als Leitfaden für die Herstellung von Sondereffekten gedacht sind. Es ist möglich, diese separat in den VC-20 einzugeben oder innerhalb von Programmen einzubetten. Selbstverständlich kann hier nicht das gesamte Repertoire des VC-20 aufgelistet werden. Der Benutzer sollte seiner Kreativität freien Lauf lassen.

Die hier aufgelisteten Spezialeffekte werden das Programm in seiner Ausführung so lange unterbrechen, bis diese vollkommen abgelaufen sind. Es ist möglich, diese Spezialeffekte in einer besonderen Art und Weise einzubauen, so daß sie die eventuell gerade laufende Zeichentrick-Darstellung nicht unterbrechen. Dieses Thema wird in allen Einzelheiten im VC-20 Programmierer-Handbuch erklärt.

Es ist hierbei zu bemerken, daß die Zeilennummern zu definieren sind, wenn diese Routinen eingegeben werden. Diese Zeilennummern

werden an dieser Stelle jedoch nicht angezeigt, um keine Verwirrung zu erzeugen, wenn es beabsichtigt ist, diese Routinen in ein schon bestehendes Programm einzubetten.

#1: SCALES

```
POKE 36878,15
FOR L=250 TO 200 STEP -2
POKE 36876,L
FOR M=1 TO 100
NEXT M
NEXT L
FOR L=205 TO 250 STEP 2
POKE 36876,L
FOR M=1 TO 100
NEXT M
NEXT L
POKE 36876,0
POKE 36878,0
```

#2: COMPUTER MANIA

```
POKE 36878,15
FOR L=1 TO 100
POKE 36876,INT(RND(1)*128)+128
FOR M=1 TO 10
NEXT M
NEXT L
POKE 36876,0
POKE 36878,0
```

#3: EXPLOSION

```
POKE 36877,220
FOR L=15 TO 0 STEP -1
POKE 36878,L
FOR M=1 TO 300
NEXT M
NEXT L
POKE 36877,0
POKE 36878,0
```

#4: BOMBS AWAY

```
POKE 36878,10
FOR L=230 TO 128 STEP -1
POKE 36876,L
FOR M=1 TO 20
NEXT M
NEXT L
POKE 36876,0
POKE 36877,200
FOR L=15 TO 0 STEP -.05
POKE 36878,L
NEXT L
POKE 36877,0
```

#5: RED ALERT

```
POKE 36878,15
FOR L=1 TO 10
FOR M=180 TO 235 STEP 2
POKE 36876,M
FOR N=1 TO 10
NEXT N
NEXT M
POKE 36876,0
FOR M=1 TO 100
NEXT M
NEXT L
POKE 36878,0
```

#6: LASER BEAM

```
POKE 36878,15
FOR L=1 TO 30
FOR M=250 TO 240 STEP -1
POKE 36876,M
NEXT M
FOR M=240 TO 250
POKE 36876,M
NEXT M
POKE 36876,0
NEXT L
POKE 36878,0
```

#7: HIGH-LOW SIREN

```
POKE 36878,15
FOR L=1 TO 10
POKE 36875,200
FOR M=1 TO 500
NEXT M
POKE 36875,0
POKE 36876,200
FOR M=1 TO 500
NEXT M
POKE 36876,0
NEXT L
POKE 36878,0
```

#8: BUSY SIGNAL

```
POKE 36878,15
FOR L=1 TO 15
POKE 36876,160
FOR M=1 TO 400
NEXT M
POKE 36876,0
FOR M=1 TO 400
NEXT M
NEXT L
POKE 36878,0
```

#9: PHONE RINGING

```
POKE 36878,15
FOR L=1 TO 5
FOR M=1 TO 50
POKE 36876,220
FOR N=1 TO 5
NEXT N
POKE 36876,0
NEXT M
FOR M=1 TO 3000
NEXT M
NEXT L
POKE 36878,0
```

#10: BIRDS CHIRPING

```
POKE 36878,15
FOR L=1 TO 20
FOR M=254 TO 240+INT (RND(1)*10) STEP -1
POKE 36876,M
NEXT M
POKE 36876,0
FOR M=0 TO INT(RND(1)*100)+120
NEXT M
NEXT L
```

#11: WIND

```
POKE 36878,15
POKE 36874,170
POKE 36877,240
FOR L=1 TO 2000
NEXT L
POKE 36874,0
POKE 36877,0
POKE 36878,0
```

#12: OCEAN WAVES

```
POKE 36877,180
FOR L=1 TO 10
D=INT(RND(1)*5)*50+50
FOR M=3 TO 15
POKE 36878,M
FOR N=1 TO D
NEXT N
NEXT M
FOR M=15 TO 3 STEP -1
POKE 36878,M
FOR N=1 TO D
NEXT N
NEXT M
NEXT L
POKE 36878,0
POKE 36877,0
```

#13: VANISHING UFO

```
POKE 36878,15
FOR L=130 TO 254
POKE 36876,L
FOR M=1 TO 40
NEXT M
NEXT L
POKE 36878,0
POKE 36876,0
```

#14: UFO LANDING

```
POKE 36878,15
FOR L=1 TO 20
FOR M=220-L TO 160-L STEP -4
POKE 36876,M
NEXT M
FOR M=160-L TO 220-L STEP 4
POKE 36876,M
NEXT M
NEXT L
POKE 36878,0
POKE 36876,0
```

#15: UFO SHOOTING

```
POKE 36878,15
FOR L=1 TO 15
FOR M=200 TO 220+L*2
POKE 36876,M
NEXT M
NEXT L
POKE 36878,0
POKE 36876,0
```

#16: WOLF WHISTLE

```
POKE 36878,15
FOR L=148 TO 220 STEP .7
POKE 36876,L
NEXT L
FOR L=128 TO 200
POKE 36876,L
NEXT L
FOR L=200 TO 128 STEP -1
POKE 36876,L
NEXT L
POKE 36878,0
POKE 36876,0
```

#17: RUNNING FEET

```
POKE 36878,15
FOR L=1 TO 10
POKE 36874,200
FOR M=1 TO 10
NEXT M
POKE 36874,0
FOR M=1 TO 100
NEXT M
NEXT L
POKE 36878,0
```

#18: TICK—TOCK

```
POKE 36878,15
FOR L=1 TO 10
POKE 36875,200
FOR M=1 TO 10
NEXT M
POKE 36875,0
FOR M=1 TO 300
NEXT M
POKE 36874,200
FOR M=1 TO 10
NEXT M
POKE 36874,0
FOR M=1 TO 300
NEXT M
NEXT L
POKE 36878,0
```

#19: DOOR OPENING

```
POKE 36878,15
B=0
FORL=128 TO 255 STEP 11
POKE 36874,L
FOR M=1 TO 10
NEXT M
B=B+1
IF B=3 THEN B=0: POKE 36874,0
NEXT L
POKE 36874,0
POKE 36878,0
```

#20: BLIPS

```
POKE 36878,15
POKE 36876,220
FOR L=1 TO 5
NEXT L
POKE 36876,0
FOR L=1 TO 500
NEXT L
POKE 36876,200
FOR L=1 TO 5
NEXT L
POKE 36876,0
FOR L=1 TO 500
NEXT L
POKE 36878,0
```

Anhang H: Codierung des Zeichenvorrats

Die folgende Tabelle listet alle Zeichen des VC-20-Zeichenvorrats. Sie zeigt die Zahlen, bzw. die Codierungen, die unter Verwendung von POKE in den Speicherbereich des Bildschirms (Positionen 7680 bis 8185) gebracht werden müssen, um das gewünschte Zeichen auf dem Bildschirm sehen zu können. Die Tabelle zeigt auch, welches Zeichen der unter Verwendung von PEEK gefundenen Zahl entspricht.

Zwar sind zwei Zeichenvorräte verfügbar, diese lassen sich jedoch unter keinen Umständen mischen. Das bedeutet, daß es nicht möglich ist, Zeichen eines bestimmten Zeichenvorrats auf dem Bildschirm anzuzeigen, wenn zur gleichen Zeit Zeichen des anderen Zeichenvorrats schon angezeigt sind. Der Wechsel von einem Zeichenvorrat zum anderen erfolgt durch die Betätigung der Tastenkombination SHIFT und COMMODORE. Dadurch wird im 2er Bit der Hauptspeicher-Adresse 36869 eine Änderung verursacht, die bedeutet, daß die Anweisung POKE 36869,240 den Zeichenvorrat auf Großbuchstaben umschaltet und POKE 36869,242 auf Kleinbuchstaben überwechselt.

Sollte sich der Benutzer ernsthaft mit Fragen der Zeichentrick-Darstellung befassen wollen, dann wird er schnell erkennen, daß die „Zeichnungen" besser gesteuert werden können, wenn die entsprechenden Zeichen unter Verwendung von POKE in den für den Bildschirm reservierten Hauptspeicher-Bereich direkt eingebracht werden (und gelöscht werden, indem der Wert 32 – Code für Leerstelle – in die gleiche Hauptspeicher-Position gebracht wird). Die Alternative dazu (jedoch die etwas umständlichere) besteht in der PRINT-Anweisung, wobei die Cursor-Steuerzeichen eingesetzt werden müssen.

Jedes in der Tabelle erscheinende Zeichen kann auch in negativer Bildschirm-Darstellung angezeigt werden. Diese Zeichen sind hier nicht aufgeführt. Das ist jedoch kein Problem, da lediglich der Wert 128 der jeweiligen Zahl hinzuaddiert werden muß.

ANMERKUNG: SIEHE AUCH BILDSCHIRM-HAUPTSPEICHERPLATZ-AUFTEILUNG

Soll z. B. ein Herzchen in Bildschirm-Position 7800 gebracht werden, muß dazu die entsprechende Nummer des Sonderzeichens in der Tabelle nachgesehen werden – in diesem Fall die Zahl 83 –, wonach der die entsprechende Bildschirm-Position (7800) ansprechende POKE-Befehl mit der Zahl für das ausgewählte Symbol (83) eingegeben werden kann.

POKE 7800,83

Ein weißes Herz sollte im mittleren Teil des Bildschirms erscheinen. Es sollte hierbei nicht vergessen werden, daß dieses Herzchen unsichtbar ist, wenn ein weißer Bildschirm-Hintergrund ausgewählt wurde. Es ist zu empfehlen, an dieser Stelle etwas zu experimentieren, indem entweder die Position geändert (größere Zahl) oder ein anderes Symbol ausgewählt wird (kleinere Zahl).

Sollte auch die Farbe des Symbols geändert werden, dann ist es notwendig, sich die beiliegende Tabelle anzusehen, in der die Farbenkodierungen einer jeden Hauptspeicherstelle aufgelistet sind. Das bedeutet selbstverständlich, daß ein zweiter POKE-Befehl benötigt wird, wenn in einer bestimmten Stelle ein bestimmtes Zeichen in anderer Farbe auszugeben ist.

Um z. B. ein rotes Herz auf dem Bildschirm erscheinen zu lassen, ist die folgende Eingabe auszuführen:

POKE 58520,2

In Bildschirm-POKEs ist der Wert der Farbe um 1 niedriger als die Ziffern der entsprechenden Farbtasten.

Der Befehl ändert die Farbe des jeweiligen Zeichens in Position 7800 auf rot. Es ist möglich, jedes Zeichen mit jeder verfügbaren Farbe zu kombinieren, indem die zwei Tabellen gedanklich zusammengebracht werden. Diese POKE-Befehle können jedem Programm hinzugefügt werden, was besonders die Zeichentrick-Darstellung belebt. Darüber hinaus ist es auch möglich, einen PEEK-Befehl auf eine bestimmte Stelle des Hauptspeichers abzusetzen, was besonders im Hinblick auf die fortgeschrittene Programmierung zum Einsatz kommt, z. B. um einen Ball springen zu lassen.

Bildschirm-Codes

SET 1	SET 2	POKE	SET 1	SET 2	POKE	SET 1	SET 2	POKE
@		0	U	u	21	*		42
A	a	1	V	v	22	+		43
B	b	2	W	w	23	,		44
C	c	3	X	x	24	—		45
D	d	4	Y	y	25	.		46
E	e	5	Z	z	26	/		47
F	f	6	[		27	Ø		48
G	g	7	£		28	1		49
H	h	8	]		29	2		50
I	i	9	↑		30	3		51
J	j	10	←		31	4		52
K	k	11	SPACE		32	5		53
L	l	12	!		33	6		54
M	m	13	“		34	7		55
N	n	14	#		35	8		56
O	o	15	$		36	9		57
P	p	16	%		37	:		58
Q	q	17	&		38	;		59
R	r	18	’		39	<		60
S	s	19	(		40	=		61
T	t	20	)		41	>		62

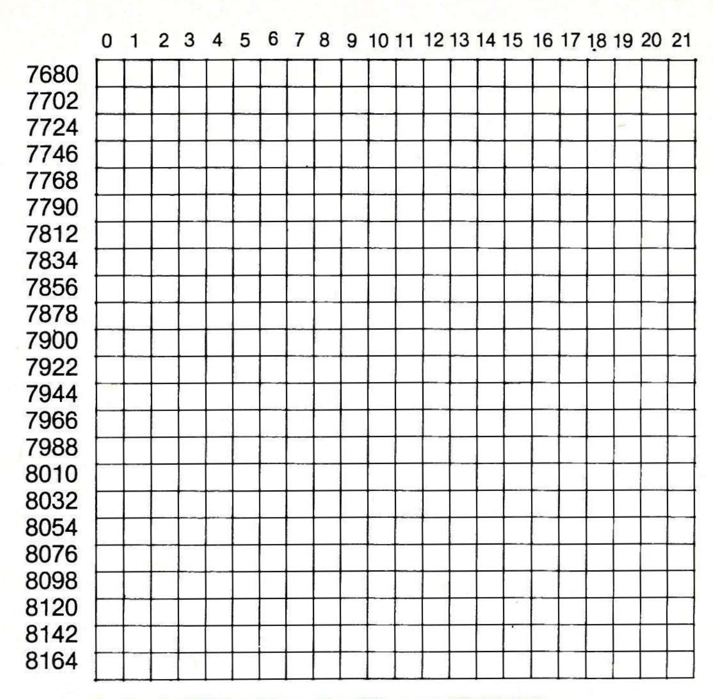
SET 1	SET 2	POKE	SET 1	SET 2	POKE	SET 1	SET 2	POKE
?		63		T	84			106
		64		U	85			107
	A	65		V	86			108
	B	66		W	87			109
	C	67		X	88			110
	D	68		Y	89			111
	E	69		Z	90			112
	F	70			91			113
	G	71			92			114
	H	72			93			115
	I	73	π		94			116
	J	74			95			117
	K	75	SPACE		96			118
	L	76			97			119
	M	77			98			120
	N	78			99			121
	O	79			100		✓	122
	P	80			101			123
	Q	81			102			124
	R	82			103			125
	S	83			104			126
					105			127

Anhang I: Hauptspeicher-Aufteilung für den Bildschirm

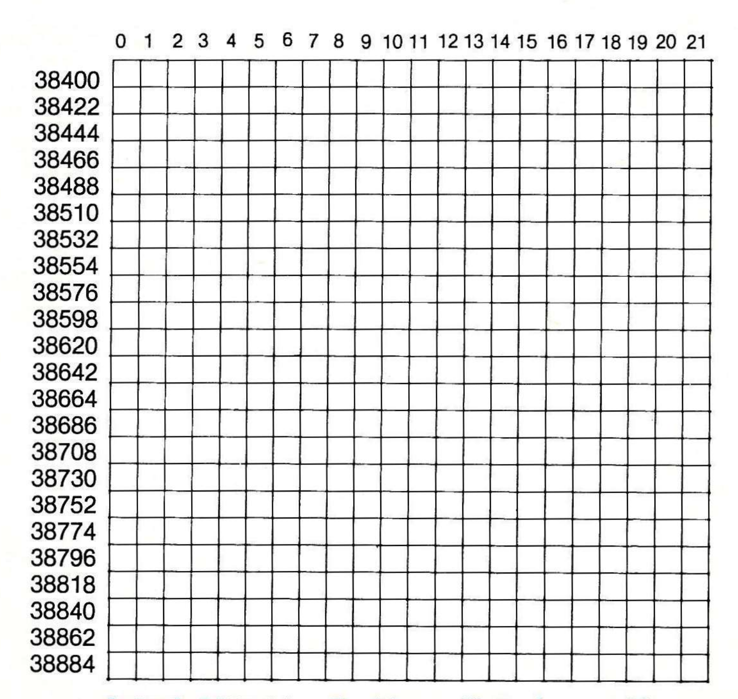
Dieser Anhang ist dazu gedacht, bei der Errechnung einer jeden Hauptspeicher-Position relativ zur Bildschirm-Position behilflich zu sein. Dazu ist lediglich die entsprechende Stelle auf dem Bildschirm zu markieren und die zwei damit im Zusammenhang stehenden Zahlen (links Zeile; oben Spalte) zu addieren. Wird es z. B. gewünscht unter Verwendung des POKE-Befehls einen „Ball" in die Mitte des Bildschirms zu bringen, dann sind die zwei Zahlen 7900 (11. Zeile) und 10 (11. Spalte) zu addieren, mit dem Ergebnis 7910. Würde der Code für den „Ball" (81, siehe Anhang H) in die Hauptspeicher-Position 7910 gebracht (POKE 7910,81), dann erscheint ein weißer Ball auf dem Bildschirm. Soll die Farbe des Balles (oder eines jeden anderen Zeichens) geändert werden, dann ist die entsprechende Bildschirm-Position des Farben-Layouts zu finden und die gleiche Addition wie auch schon vorher auszuführen, jedoch mit anderen Werten (38620 + 10 = 38630). Da jetzt die entsprechende Hauptspeicher-Position für den Farbenwechsel bekannt ist, muß lediglich die gewünschte Farbe ausgewählt werden. Der abzusetzende POKE-Befehl lautet in diesem Fall POKE 38630,3 wobei für dieses Beispiel angenommen wurde, daß die Farbe türkis sein soll. Es ist besonders darauf zu achten, daß unter Verwendung der POKE-Anweisung der Wert für die ausgewählte Farbe um 1 kleiner ist, als die Ziffern der Tasten, die in Verbindung mit der jeweils selektierten Farbe stehen (siehe unten).

Kurzauflistung der Farbcodierungen:

Code	Farbe
0	Schwarz
1	Weiß
2	Rot
3	Türkis
4	Violett
5	Grün
6	Blau
7	Gelb



Seite 1: Bildschirm-Positionen für Zeichen



Seite 2: Bildschirm-Positionen für Farbauswahl

Anhang J: ASCII- und CHR$-Code

Dieser Anhang zeigt an, welche Zeichen auf dem Bildschirm erscheinen, wenn unter Verwendung von PRINT CHR$(X) alle möglichen Werte von X ausgegeben werden sollen. Es werden auch die Werte gezeigt, die durch die Eingabe von PRINT ASC(„X") mit X in Verbindung stehen. X kann jedes auf der Tastatur verfügbare Zeichen aufnehmen. Diese Einrichtung dient dem Überblick der empfangenen Zeichen, wenn eine GET-Anweisung ausgeführt wird, welche Groß-/Kleinbuchstaben konvertiert und zeichenbezogene Befehle ausgibt (wie Wechsel zu Groß-/Kleinbuchstaben (was nicht in Anführungszeichen eingebettet werden konnte).

PRINTS	CHR$	PRINTS	CHR$	PRINTS	CHR$	PRINTS	CHR$
	0		16	SPACE	32	Ø	48
	1	CRSR ↓	17	!	33	1	49
	2	RVS ON	18	"	34	2	50
	3	CLR HOME	19	#	35	3	51
	4	INST DEL	20	$	36	4	52
WHT	5		21	%	37	5	53
	6		22	&	38	6	54
	7		23	.	39	7	55
	8		24	(	40	8	56
	9		25	)	41	9	57
	10		26	*	42	:	58
	11		27	+	43	;	59
	12	RED	28	,	44	<	60
RETURN	13	CRSR ⇒	29	—	45	=	61
SWITCH TO LOWER CASE	14	GRN	30	.	46	>	62
	15	BLU	31	/	47	?	63

PRINTS	CHR$	PRINTS	CHR$	PRINTS	CHR$	PRINTS	CHR$
@	64	U	85		106		127
A	65	V	86		107		128
B	66	W	87		108		129
C	67	X	88		109		130
D	68	Y	89		110		131
E	69	Z	90		111		132
F	70	[	91		112	f1	133
G	71	£	92		113	f3	134
H	72	]	93		114	f5	135
I	73	↑	94		115	f7	136
J	74	←	95		111	f2	137
K	75		96		117	f4	138
I	76		97		118	f6	139
M	77		98		119	f8	140
N	78		99		120	SHIFT RETURN	141
O	79		100		121	SWITCH TO UPPER CASE	142
P	80		101		122		143
Q	81		102		123	BLK	144
R	82		103		124	↑ CRSR	145
S	83		104		125	RVS OFF	146
T	84		105	π	126	CLR HOME	147

PRINTS	CHR$	PRINTS	CHR$	PRINTS	CHR$	PRINTS	CHR$
INST DEL	148	CYN	159		170		181
	149	SPACE	160		171		182
	150		161		172		183
	151		162		173		184
	152		163		174		185
	153		164		175		186
	154		165		176		187
	155		166		177		188
PUR	156		167		178		189
⇐ CRSR	157		168		179		190
YEL	158		169		180		191

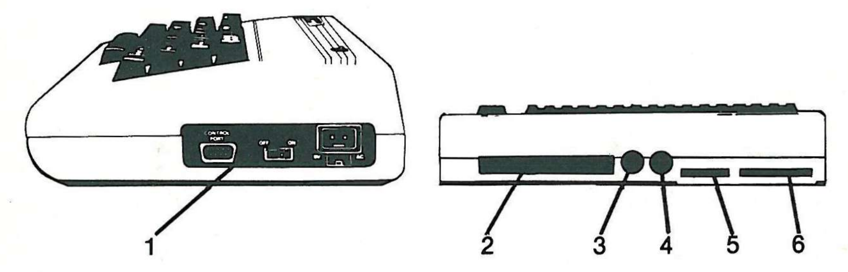
Anhang K: Ableitung mathematischer Funktionen

Mathematische Funktionen, die nicht durch die BASIC-Programmiersprache des VC-20 erhältlich sind, können folgendermaßen abgeleitet werden:

FUNKTION	VC-20 BASIC-SPRACHE
SECANT	SEC(X) = 1/COS(X)
COSECANT	CSC(X) = 1/SIN(X)
COTANGENT	COT(X) = 1/TAN(X)
INVERSE SINE	ARCSIN(X) = ATN(X/SQR(– X*X + 1))
INVERSE COSINE	ARCCOS(X) = – ATN(X/SQR (– X*X + 1)) + π/2
INVERSE SECANT	ARCSEC(X) = ATN(X/SQR(X*X – 1))
INVERSE COSECANT	ARCCSC(X) = ATN(X/SQR(X*X – 1)) + (SGN(X) – 1)* π/2
INVERSE COTANGENT	ARCOT(X) = ATN(X) + π/2
HYPERBOLIC SINE	SINE(X) = (EXP(X) – EXP(– X))/2
HYPERBOLIC COSINE	COSH(X) = (EXP(X) + EXP(– X))/2
HYPERBOLIC TANGENT	TANH(X) = EXP(– X)/EXP(X) + EXP (– X))*2 + 1
HYPERBOLIC SECANT	SECH(X) = 2/(EXP(X) + EXP(– X)
HYPERBOLIC COSECANT	CSCH(X) = 2/(EXP(X) – EXP(– X))
HYPERBOLIC COTANGENT	COTH(X) = EXP(– X)/(EXP(X) – EXP(– X)*2 + 1
INVERSE HYPERBOLIC SINE	ARCSINH(X) = LOG(X + SQR(X*X + 1)
INVERSE HYPERBOLIC COSINE	ARCCOSH(X) = LOG(X + SQR(X*X – 1)
INVERSE HYPERBOLIC TANGENT	ARCTANH(X) = LOG((1 + X)/(1 – X))/2
INVERSE HYPERBOLIC SECANT	ARCSECH(X) = LOG((SQR (– X*X + 1) + 1)/X)
INVERSE HYPERBOLIC COSECANT	ARCCSCH(X) = LOG((SGN(X)*SQR (X*X + 1)/X
INVERSE HYPERBOLIC COTANGENT	ARCCOTH(X) = LOG((X + 1)/(X – 1))/2

Anhang L:
Anschlüsse für Eingabe-/Ausgabe-Geräte

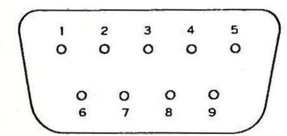
Es folgt eine Abbildung der Anschlüsse hinsichtlich der Eingabe-/Ausgabe-Geräte des VC-20:



1. Eingabe/Ausgabe für Spiele
2. Hauptspeicher-Erweiterung
3. Audio und Video
4. Serieller Eingabe-/Ausgabe-Anschluß (Diskette)
5. Kassette
6. Anwenderanschluß (Modem)

---

1. Eingabe/Ausgabe für Spiele

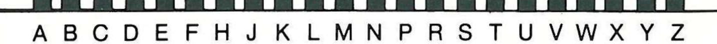


PIN #	TYP	BEMERKUNG
1	JOYØ	
2	JOY1	
3	JOY2	
4	JOY3	
5	POT Y	
6	LIGHT PEN	
7	+5V	MAX. 1ØØmA
8	GND	
9	POT X	

Joystick Schalter gegen GND

Pot: +5 POT

2. Hauptspeicher-Erweiterung



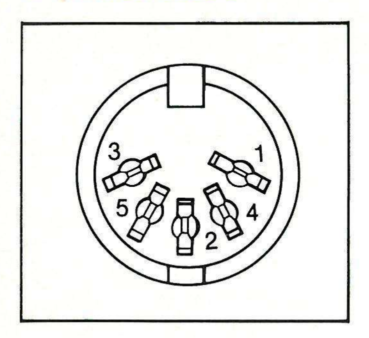
PIN #	TYP
1	GND
2	CDØ
3	CD1
4	CD2
5	CD3
6	CD4
7	CD5
8	CD6
9	CD7
1Ø	$\overline{\text{BLK1}}$
11	$\overline{\text{BLK2}}$

PIN #	TYP
12	$\overline{\text{BLK3}}$
13	$\overline{\text{BLK5}}$
14	$\overline{\text{RAM1}}$
15	$\overline{\text{RAM2}}$
16	$\overline{\text{RAM3}}$
17	VR/W
18	CR/W
19	$\overline{\text{IRQ}}$
2Ø	NC
21	+5V
22	GND

PIN #	TYP
A	GND
B	CAØ
C	CA1
D	CA2
E	CA3
F	CA4
H	CA5
J	CA6
K	CA7
L	CA8
M	CA9

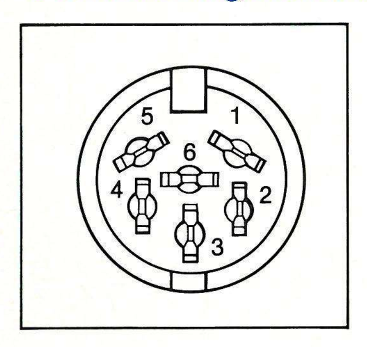
PIN #	TYP
N	CA1Ø
P	CA11
R	CA12
S	CA13
T	I/Ø2
U	I/Ø3
V	SØ2
W	$\overline{\text{NMI}}$
X	$\overline{\text{RESET}}$
Y	NC
Z	GND

3. Audio/Video



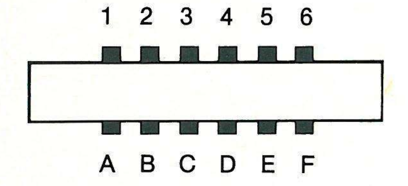
PIN #	TYP	BEMERKUNG
1	+6V	10mA MAX
2	GND	
3	AUDIO	
4	VIDEO LOW	
5	VIDEO HIGH	

4. Serielle Eingabe/Ausgabe



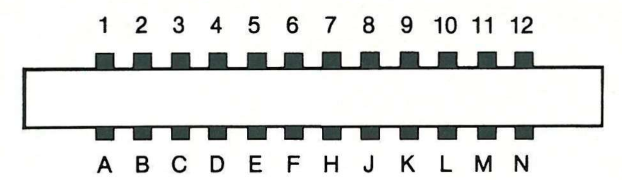
PIN #	TYP
1	SERIAL SRQ IN
2	GND
3	SERIAL ATN IN/OUT
4	SERIAL CLK IN/OUT
5	SERIAL DATA IN/OUT
6	NC

5. Kassette



PIN #	TYP
A-1	GND
B-2	+5V
C-3	CASSETTE MOTOR
D-4	CASSETTE READ
E-5	CASSETTE WRITE
F-6	CASSETTE SWITCH

6. Anwender Eingabe/Ausgabe



PIN #	TYPE	BEMERKUNG	PIN #	TYPE	BEMERKUNG
1	GND		A	GND	
2	+5V	1ØØmA MAX.	B	CB1	
3	RESET		C	PBØ	
4	JOYØ		D	PB1	
5	JOY1		E	PB2	
6	JOY2		F	PB3	
7	LIGHT PEN		H	PB4	
8	CASSETTE SWITCH		J	PB5	
9	SERIAL ATN IN		K	PB6	
1Ø	+9V	1ØØmA MAX.	L	PB7	
11	GND		M	CB2	
12	GND		N	GND	

Anhang M: VC-20 Versuchsprogramme

```
1 REM TANK-V-UFO BY D.LATER
5 OF=30720
6 VA=9*16↑3+14:VN=9*16↑3+13
8 POKEVA+1,25
9 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒______________________";
20 PRINT"▒** TANK VERSUS UFO **"
21 PRINT"Z-LEFT,C-RIGHT,B-FIRE"
100 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
101 FORA=38400TO38400+22*23   :POKEA,0:NEXT:A=0
105 PRINT"▒    |"
110 PRINT" _▒  ▒_"
115 PRINT" |oooo| "
120 PRINT"▒▒▒";
125 IFPEEK(197)=34THENGOSUB300:A=A+1:GOTO220
130 IFPEEK(197)=33THENGOSUB300:A=A-1:GOTO220
135 IFPEEK(197)=35ANDB=0THENB=1:C=8079+A:L=1:P=15:POKEVN,158:POKEC,160:POKEC+OF
4:GOTO170
136 IFL=0THEN140
137 POKEVA,P:P=P-1:IFP=-1THENL=0:POKEVN,0
140 IFB=0THEN170
141 FORH=1TO2
145 POKEC,32:C=C-22
150 IFPEEK(C)=60ORPEEK(C)=42ORPEEK(C)=62    THEN600           :
155 IFC<7746THENB=0:GOTO170         :
160 POKEC,46
161 NEXT
```

Spezial-Symbol-Schlüssel

```
"▒" IS CLR
"▒" IS HOME
"▒" IS CRSR DOWN
"▒" IS CRSR LEFT
"▒" IS CRSR UP
"▒" IS CTRL-2
"▒" IS CTRL-3
"▒" IS CTRL-5
"▒" IS CTRL-1
```

```
170 IFD=0THEN800
172 IFD=0THEN180
174 POKEE,32:POKEE-1,32:POKEE-2,32:K=K+1
176 IFE=ITHEND=0:GOTO180
178 E=E+J: POKEE,62:POKEE-1,42:POKEE-2,60
179 IFJ=1THEN182
180 IFINT((8186-E)/22)=22-K-AANDF=0THEN F=1:G=E+21:M=21:GOTO183       :
181 GOTO183
182 IFINT((8098-E)/22)=A-KANDF=0THENF=1:G=E+23:M=23
183 IFF=0THEN125
184 POKEG,32:G=G+M
186 IFPEEK(G)<>32THEN700
187 IFG>7680+22*21THENF=0:GOTO500
189 POKEG,81:GOTO125
220 IFA<0THENA=0
221 IFA>15THENA=15
222 PRINTTAB(A)"    |"
225 PRINTTAB(A)" _■  ■_"
230 PRINTTAB(A)" |OOOO| "
235 PRINT"]]]";:GOTO135
300 PRINTTAB(A)"       "
305 PRINTTAB(A)"       "
310 PRINTTAB(A)"        "
315 PRINT"]]]";:RETURN
400 PRINTPEEK(197):GOTO400
500 POKEG,66:POKEG+1,78:POKEG-1,77:POKEG-20,46:POKEG-21,46:POKEG-22,46
510 POKEG-23,46:POKEG-24,46
520 FORAA=1TO100:NEXT
530 POKEG,32:POKEG+1,32:POKEG-1,32:POKEG-20,32:POKEG-21,32:POKEG-22,32
535 POKEG-23,32:POKEG-24,32
```

```
590 GOTO125
600 POKEC,160:POKEC+1,160:POKEC-1,160:POKEC+22,160:POKEC-22,160
601 L=0
610 POKEVN,128+100
611 FORGG=15TO0STEP-1:POKEVA,GG:FORGH=1TO70:NEXT:NEXT
615 B=0:D=0
616 POKEC,32 :POKEC+1, 32:POKEC-1, 32:POKEC+2, 32:POKEC-2,32:POKEC+3,32:POKEC-3
32
617 POKEC-22,32:POKEC+22,32
640 E=E+22+J:POKEVA,15:POKEVN,0:
645 POKEE,62:POKEE-1,42:POKEE-2,60
646 FORO=248TO253:POKEVN-1,O:NEXT:FORO=253TO248STEP-1:POKEVN-1,O:NEXT
647 POKEE,32:POKEE-1,32:POKEE-2,32
650 IFE<7680+20*22THEN640
651 E=E+J
652 POKEE+22,62:POKEE+21,42:POKEE+20,60: POKEE+OF,4:POKEE+OF-1,4:POKEE+OF-2,4
653 POKEE+22+OF,0:POKEE+21+OF,0:POKEE+20+OF,32
654 POKEVN-1,0:POKEVN,128:FORQ=1TO20:POKEVA,15-INT(Q/1.33)
655 POKEE,223:POKEE-1,223:POKEE-2,223:FORO=1TO80:NEXT
656 POKEE,233:POKEE-1,233:POKEE-2,233:FORO=1TO80:NEXT
657 NEXT:POKEE,32:POKEE-1,32:POKEE-2,32:POKEE+22,32:POKEE+21,32:POKEE+20,32 :
658 PRINT"                    "
659 DU=DU+1:PRINT"UFOS"DU"TANKS"DT:PRINT""
660 GOTO125
700 POKEVN,128:L=0
701 A=A+1: FORKL=1TO200:POKEVA,15-INT(KL/13):
704 PRINTTAB(A)""
705 PRINTTAB(A)"      "
715 PRINT"";
720 PRINTTAB(A)""
```

```
725 PRINTTAB(A)"       "
735 PRINT"";
740 NEXT
745 PRINTTAB(A)"    "
750 PRINTTAB(A)"    "
751 PRINTTAB(A-1)"       ";
752 PRINT"";
756 PRINT"              ;
752 PRINT"";
756 PRINT"                   "
757 DT=DT+1:PRINT"UFOS"DU"TANKS"DT:PRINT""
760 F=0:  A=0:GOTO105
800 D=1:E=7702+INT(RND(1)*14)*22+88  :I=E-20:K=0:J=-1:IFRND(1)>.5THENE=E-21:I=E+
20:J=1
805 GOTO172
```

```
0 REM""    KILLER COMET        BY  DUANE LATER
1 C=7680+22*6:DIMB(12):LL=C:POKE9*16↑3+15,8
2 PRINT"*** KILLER COMET  ***":T=0
3 PRINT"     HIT ANY KEY"
8 POKE8179,160:
9 FORA=38444TO38400+505:POKEA,1:NEXT
10 :
20 FORA=1 TO 12:B(A)=160:NEXT
21 IFW=0THEN25
22 POKE9*16↑3+14,U:U=U-2:IFU<=0THENW=0:POKE9*16↑3+13,0
25 F=1:REM ERA METEOR
```

```
 26 FORE=0TO44STEP22
 27 FORD=C+ETOC+3+E:POKED,32  :F=F+1:NEXT:NEXT
 28 C=C+1
 29 F=1:REM DRAW METEOR
 30 POKE8179,160:FORE=0TO44STEP22
 35 FORD=C+ETOC+3+E:POKED,B(F):F=F+1:NEXT:NEXT
 36 IFPEEK(8178)=160THENPRINT"[illegible]MOON BASE DESTROYED !":GOTO500
 40 GETA$:IFA$<>""ANDG=0THENG=1:S=7680+15+22*21
 50 IFG=0THEN80
 55 POKES,32:S=S-22
 60 IFS<7746THENG=0:GOTO21
 70 IFPEEK(S)=160THENPOKES,32:G=0:T=T+1:W=1:POKE9*16↑3+13,128+000:U=15:GOTO80
 71 IFPEEK(S-1)=160THENG=0:POKES-1,32:T=T+1:W=1:POKE9*16↑3+13,128+000:U=15:GOTO8
0
 75 POKES,81
 80 F=1:REM CHECK MET
 81 IFT=12THENPRINT"[illegible]**METEOR DESTROYED***":FORRR=1TO2500:NEXT:LL=LL+44:C=LL
:GOTO2
 82 FORE=0TO44STEP22
 84 FORD=C+ETOC+3+E:IFPEEK(D)=32THENB(F)=32
 86 F=F+1:NEXT:NEXT
 90 GOTO21
 500 POKE9*16↑3+13,128+5
 505 POKE9*16↑3+14,5:FORRR=1TO300:NEXT
 510 FORA=15TO0STEP-1
 511 POKE9*16↑3+14,A
 520 FORRR=1TO500:NEXT
 530 NEXT
 540 FORRR=1TO2000:NEXT:RUN
READY.
```

```
1 REM"████████"
2 REM"█████████"    ROCKET COMMAND        BY  DUANE LATER
10 VI=9*16↑3:OF=38400-7680:PRINT"█"
11 FORA=38400+22TO38400+22*23:POKEA,0:NEXT
15 C=7680+22*20+15
20 POKEVI+15,6+128+64+32+8
30 PRINT"██*** ROCKET COMMAND ***";
35 PRINT"      HIT ANY KEY"
40 PRINT"████████████████████████████████████████████";
50 PRINT"██████████████████↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑"
60 A=7680+22*22+15
70 GET A$:IFA$<>""ANDB=0THENB=1:POKEC+22,32:D=C:C=C-1:K=1:POKE9*16↑3+13,128+125
:L=16
71 IFK=1THENL=L-1:POKE9*16↑3+14,L
72 IFL=0THENK=0:POKE9*16↑3+13,0
75 IFC=8121THENPRINT"████████NO MORE ROCKETS!!!":POKE9*16↑3+14,0: FORAA=1TO3000:
NEXT:RUN
80 IFB=0THEN110
85 POKED,32:D=D-22
90 IFPEEK(D)=60ORPEEK(D)=62THENGOSUB500:GOTO70
100 IFD<7680+88THENB=0:GOTO110
105 POKED,30
110 IFH=0THEN200
115 IFH=0THEN70
120 POKEF,32:F=F+E
125 IFF=ITHENH=0:GOTO70
130 IFPEEK(F)=30THENGOSUB500:GOTO70
140 POKEF,G:GOTO70
200 H=1
205 IFINT(RND(1)*2)=1THENE=-1:F=7702+(INT(RND(1)*10)+6)*22:I=F-22:G=60:  GOTO11
5
```

```
210 E=1:F=7680+(INT(RND(1)*10)+6)*22:I=F+22:G=62: GOTO115
500 B=0:H=0
501 SC=SC+10:PRINT"[illegible]SCORE = "SC
502 POKEF+OF,4:POKEF+1+OF,4:POKEF-1+OF,4
503 POKEF+OF+22,4:POKEF-22+OF,4
510 POKEF,160:POKEF+1,160:POKEF-1,160:POKEF+22,160:POKEF-22,160
521 POKE9*16↑3+13,128+35
522 FORY=16TO0STEP-1
523 POKE9*16↑3+14,Y
524 FORP=1TO80:NEXT:NEXT
530 POKEF,32:POKEF+1,32:POKEF-1,32:POKEF+22,32:POKEF-22,32
531 POKEF+OF,0:POKEF+1+OF,0:POKEF-1+OF,0
532 POKEF+OF+22,0:POKEF-22+OF,0
533 POKE9*16↑3+13,0
540 FORGH=FTOF+22*16STEP22
544 II=PEEK(GH):POKEGH,G:FOROO=1TO60:NEXT
546 POKEGH,II:NEXT
800 RETURN
999 GOTO70
1000 POKE9*16↑3+13,128+125
1001 FORY=16TO0STEP-1
1005 POKE9*16↑3+14,Y
1010 NEXT:POKE9*16↑3+13,0
1020 RETURN
READY.
```

Anhang N: Fehlermeldungen

Dieser Anhang enthält die vollständige Liste der von dem VC-20 generierten Fehlermeldungen, einschließlich deren Ursachen und einer Kurzbeschreibung.

BAD DATA (UNGÜLTIGE EINGABEWERTE)... Es wurden alphanumerische Zeichen empfangen, obwohl rein numerische Daten erwartet wurden.

BAD SUBSCRIPT (UNGÜLTIGE INDIZIERUNG)... Es wurde in dem Programm versucht, ein Element in einer Tabelle anzusprechen, dessen Adresse sich außerhalb der mittels der DIM-Anweisung spezifizierten Bandbreite befindet.

CAN'T CONTINUE (FORTSETZUNG NICHT MÖGLICH)... Der CONT-Befehl ist nicht einsetzbar, da das Programm nie mit RUN ausgeführt wurde, ein Fehler aufgetreten ist oder eine Programmzeile geändert wurde.

DEVICE NOT PRESENT (PERIPHERIE-GERÄT NICHT VERFÜGBAR)... Ein angesprochenes Eingabe-/Ausgabe-Gerät stand für einen der folgenden Befehle nicht zur Verfügung: OPEN, CLOSE, CMD, PRINT#, INPUT#, GET#, SAVE, LOAD, VERIFY.

DIVISION BY ZERO (DIVISION DURCH NULL)... Die Division durch Null ist mathematisch nicht ausführbar und deswegen auch nicht erlaubt.

EXTRA IGNORED (ZUSÄTZLICHE EINGABEN ÜBERGANGEN)... Als Antwort zu einer INPUT-Anweisung wurden zu viele Daten eingegeben. Es wurden nur einige der ersten Elemente angenommen. Eingaben von Zeichenketten die ein Komma beinhalten, werden nur bis zum Komma gespeichert, dann erfolgt obige Meldung ohne Programmstop.

FILE NOT FOUND (DATEI NICHT GEFUNDEN)... Diese Nachricht erscheint, wenn entweder eine Datei auf Magnetband gesucht und Bandende-Kennzeichen gefunden wurde oder, wenn der Name einer entsprechenden Datei auf Diskette nicht vorrätig ist.

FILE NOT OPEN (DATEI NICHT ERÖFFNET)... Die mit den Befehlen CLOSE; CMD, PRINT#, INPUT# oder GET# angesprochene Datei muß zuerst mit dem Befehl OPEN eröffnet werden.

FILE OPEN (DATEI BEREITS ERÖFFNET)... Es wurde versucht eine Datei mit einer Nummer zu eröffnen, die bereits eröffnet ist.

FORMULA TOO COMPLEX (FORMEL ZU KOMPLEX)... Die hier analysierte Formel sollte in zumindest zwei Teile aufgeteilt werden, damit das System die entsprechenden Anweisungen ausführen kann.

ILLEGAL DIRECT (UNGÜLTIG IM DIREKTMODUS)... Die INPUT-Anweisung kann nur in Verbindung mit einem Programm eingesetzt werden und nicht im Direktmodus.

ILLEGAL QUANTITY (UNGÜLTIGER WERT)... Eine in einer Bedingung, einer Funktion oder einer Anweisung verwendete Zahl steht außerhalb der erlaubten Bandbreite.

LOAD (LADEN)... Ein Problem hat sich im Zusammenhang mit dem Laden von Band ergeben.

NEXT WITHOUT FOR (NEXT OHNE FOR)... Diese Nachricht wird entweder durch eine ungültige Verschachtelung von Programmschleifen oder durch die Benutzung eines Datenfeldnamens in einer NEXT-Anweisung verursacht, die nicht mit einer der FOR-Anweisungen korrespondiert.

NOT INPUT FILE (KEINE EINGABEDATEI) . . . Es wurde versucht, eine Datei mit den Befehlen INPUT oder GET anzusprechen, obwohl diese Datei als Ausgabedatei definiert wurde.

NOT OUTPUT FILE (KEINE AUSGABEDATEI) . . . Es wurde versucht, mittels des Befehls PRINT Daten auf einer Datei zu speichern, welche als Eingabedatei deklariert wurde.

OUT OF DATA (KEINE DATA-ANWEISUNGEN ÜBRIG) . . . Es wurde eine READ-Anweisung ausgeführt, obwohl über DATA-Anweisungen keine Daten mehr zur Verfügung stehen.

OUT OF MEMORY (HAUPTSPEICHER-ÜBERLAUF) . . . Für das Programm oder dessen Daten ist die Hauptspeicher-Kapazität nicht ausreichend. Diese Nachricht kann auch dann abgesetzt werden, wenn zu viele FOR-Schleifen verschachtelt wurden oder wenn zu viele GOSUBs in Kraft sind.

OVERFLOW (ÜBERLAUF) . . . Das Ergebnis einer Kalkulation ist größer als erlaubt (größer als 1.70141884E+38).

REDIM'D ARRAY (ERNEUTES DIM FÜR FELDER) . . . Die Anweisung DIM kann im Zusammenhang mit einer Tabelle nur ein Mal angewendet werden. Wird ein Element einer Tabelle vor Ausführung der DIM-Anweisung angesprochen, erfolgt eine automatische DIM-Operation auf die entsprechende Tabelle, wobei standardgemäß 10 Elemente eingerichtet werden. Jede nun darauf folgende definierte DIM-Anweisung verursacht diesen Fehler.

REDO FROM START (ERNEUTE EINGABE VOM ANFANG) . . . Als Antwort zu einer INPUT-Anweisung wurde die Eingabe von numerischen Zeichen erwartet, jedoch alpha-numerische Zeichen eingegeben. Die Information ist lediglich nochmals vom Anfang an einzugeben, wonach das Programm automatisch fortgesetzt wird.

RETURN WITHOUT GOSUB (RETURN OHNE GOSUB) . . . Es wurde eine RETURN-Anweisung gefunden und keine GOSUB-Anweisung abgesetzt.

STRING TOO LONG (ZEICHENKETTE ZU LANG) . . . Eine alpha-numerische Zeichenkette kann bis zu 255 Zeichen enthalten.

SYNTAX . . . Der VC-20 erkennt eine abgesetzte Anweisung nicht an. Dabei könnte es sich um fehlende Klammern, falsch buchstabierte Schlüsselwörter, usw. handeln.

TYPE MISMATCH (FALSCHE DATENART) . . . Dieser Fehler taucht auf, wenn anstatt eines rein numerischen Wertes alpha-numerische Zeichen angegeben wurden oder umgekehrt.

UNDEF'D FUNCTION (NICHT DEFINIERTE FUNKTION) . . . Es wurde eine anwenderdefinierte Funktion angesprochen, die jedoch niemals mit der DEF FN-Anweisung definiert wurde.

UNDEF'D STATEMENT (NICHT DEFINIERTE ANWEISUNG) . . . Es wurde versucht, sich mit einer GOTO- oder GOSUB-Anweisung oder mit RUN auf eine Programmzeile zu beziehen, die nicht besteht.

VERIFY (PRÜFUNG) . . . Das sich auf Band oder Diskette befindende Programm entspricht nicht dem sich im Hauptspeicher befindenden Programm.

STICHWORTVERZEICHNIS

V

W

Z

Öffnen: 3 Schrauben unter
Space-Taste

Videopegel: orig ca 350 mV
einstellbar mit R7

Ein „benutzerfreundlicher“ Computer

Der neue VC-20 Computer wird als der benutzerfreundlichste Computer auf dem Markt bezeichnet; freundlich im Preis, in der Größe, in der Benutzung und Erweiterung.

Mit dem VC-20 liefert Commodore ein Computersystem, welches nahezu jedem hilft, schnell und leicht mit einem Computer zu arbeiten. Weiterhin ist das System für den Anschluß von vielen verschiedenen Erweiterungen ausgelegt, so daß es mit dem Wissen und den Anforderungen des Anwenders wachsen kann.

VC-20 Besitzer, welche mehr über die Arbeit mit dem Computer lernen möchten, sollten Ihren Commodore Händler nach folgenden Lehrmitteln fragen:

- **VC-20 Lernserie ...** eine Bibliothek von Lehrbüchern und Kassetten, die Ihnen helfen, die Arbeit mit dem Computer und mit anderen Dingen zu erlernen. Die Serie beinhaltet das Arbeiten mit dem VC-20 Volkscomputer, Einführung in BASIC, Unterhaltung, Töne und Musik uvm.

- **VC-20 Programmierhandbuch ...** umfassendes Buch über den VC-20 Volkscomputer, beinhaltet wichtige Informationen zur Programmierung des VC-20, für Anfänger genauso geeignet wie für erfahrene Programmierer.

- **VC-Programmkassetten und Disks ...** eine wachsende Bibliothek von Unterhaltungs-, Ausbildungs- und Dienstprogrammen mit welchen man den VC-20 zur Lösung von Aufgaben, zum Erlernen verschiedener Dinge und zum Spielen bestehender TV-Spiele benutzen kann. Diese leicht zu bedienenden Programme erfordern keine Computererfahrung.

